FinalYearReport_SoyaShrestha_77356846.pdf

by Soya Shrestha

Submission date: 14-May-2025 01:51PM (UTC+0545)
Submission ID: 2675572190
File name: FinalYearReport_SoyaShrestha_77356846.pdf (5.89M)
Word count: 9492
Character count: 54173

Final Year Project 2025

Smart Vitals

A Portable Health Monitoring System

Submitted by

Soya Shrestha

University Id: (77356846)

BSC (Hons) Computing

Supervisor: Rohit Raj Pandey

Abstract

In todays' modern technologically driven world, Smart Vitals presents itself as an IoT enabled, multifunctional and portable device for health tracking. Designed to provide users with an affordable solution replacing traditional tools, it allows people to get insights on their health and surroundings with a single device. The main focus of developing this device is to make health readings accessible to people wherever they are. It not only saves valuable time of the users but also allows them to take preventive measures. Smart Vitals aims to promote health sector as it provides all-in-one solution for monitoring basic vitals. While Smart Vitals is still a prototype, with further enhancement, proper research and refining, it has a promising scope.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1: INTRODUCTION

Smart Vital is a portable health monitoring device. It is designed to measure vital signs of an individual. It includes $Sp0_2$ level, heart rate, ECG, temperature, and blood pressure measurement by utilizing embedded system and IoT. The device provides easy and quick access to essential health parameters in real-time through locally as well as through mobile application. The main purpose for the development of the device is to facilitate users with a reliable and user-friendly product for the healthcare management system.

The device is developed so that it can be placed at first aid or medical box at home or it can be used while travelling as it is compact and easy to carry. A single compact device is used unlike having different devices for measuring health vitals. It offering all in one general health measurement, making sure that you have all the necessary health information well within access whether you are on the go or at home. The device Smart Vitals fulfils the increase in the need and demand of the health care system as it is portable and convenient to use. The system uses ESP32 WROVER as microcontroller, which is the main processing segment for collecting the data from sensors like Max30100, AD8232, 10K NTC Thermistor temperature sensor. The readings from the sensor are locally access via OLED display and also transmitted to mobile application made via Flutter. If there is any abnormality in the reading, alert is sent, ensuring timely intervention.

Smart Vitals was compared with various existing products, during which the uniqueness of the smart vitals was discovered. Additionally, the key features of the smart vitals were also highlighted. All the hardware and software components were well-identified, even the smallest of the components. Without the use of those components the working of smart vitals would lack behind. Smart Vitals followed the agile methodology for its development as it allows flexibility and phased development. The product design includes the overall flow of the system as well as the flow of users interacts with the device and system. The features of smartvitals with the level of priority was emphasized. After the product was developed, the product was tested continuously finalizing the products'

evaluation ensuring that the system passed every test for both device and mobile application. MS Project was utilized to keep track of the progress of the project with proper planned layout, while GitHub was used for version control and to keep track of the changes made in the project. Similarly, ERD was used to get knowledge on the data structure and better understanding of the database model. At last, the overall project summary and outcome was discussed, including the performance of the system and how the system could be further enhanced for future works.

CHAPTER 2: REVIEW OF LITERATURE

The continuous advancement in technologies in health care has led to advanced approaches like Smart Vitals. The importance of products like Smart Vitals has emerged as a necessary device in healthcare and management due to the rapid global interest for its portability and remote health monitoring features. The incorporation of IoT and micro-controller enable the device to provide users with essential health vitals giving a proper and prompt insights into a person's health state providing assurance to the user. By gaining knowledge on personal health, it allows users to take an active part in maintaining their health and taking precautions as needed. As the device has features such as being portable and remotely accessibility, it makes devices like Smart Vitals to be effective and beneficial for any individual to keep track of their health, even on the areas with insufficient healthcare facilities. Not only that it provides vitals information through mobile application as well, making it easier for the users to keep track of their health.

Smart Vitals: A Portable Health Monitoring System, summarizes the importance of compact device, high-tech health monitoring system via advanced technology. It focuses on device's intelligence, accessibility with user friendly components making it advanced, accurate and efficient health measuring system made to

achieve present days' requirements. The device helps to tackle obstacles related to healthcare management by allowing users to view their real time health vitals. One of the major benefits is the ability to minimize the need for medical visits for regular health check-ups, reducing expense and saving time. In todays' modern market, people are more inclined towards smart healthcare system including wearable technologies, Smart Vital distinguishes itself by presenting value for money, live tracking, and mobile application integration.

The "Android Based Health Parameter Monitoring" by (Trivedi & Cheeran , 2017) has embedded system with Android mobile app that tracks and keeps record of vital heath signs which includes body temperature and heart rate via sensors and transmits data via Bluetooth to mobile app. While Smart Vitals offers wide range of health parameters for monitoring including ECG, blood pressure and SpO_2 level. The data transmission is real-time based and quick in the mobile application via Wi-Fi connection. The mobile application developed by (Trivedi & Cheeran , 2017) is very minimal delivering limited user interaction, with basic display of readings. In comparison, Smart Vital offers more user engagement with visual analytics. The system has dashboard which allows users to view detailed data trends, elevating user experience allowing the app to be user-friendly and helpful.

According to (Varghese & Muthukumaraswamy, 2025)'s "An IoT-Based Health Monitoring System for Elderly Patients", health care has been lacking behind a lot due to growth in elderly population in China, which is the reason for the development of remote health monitoring device. The device includes components for measuring pulse rate, blood pressure, temperature and oxygen saturation and Arduino as the core processing unit. Although the system by (Varghese & Muthukumaraswamy, 2025) offers most of what Smart Vitals offers, the system does not contain mobile application which is somewhat a necessary part as it allows users to view data anywhere with Wi-Fi connection. Smart Vital offers a feature to send alert such as voice alert through the mobile application and via Email as well with device location to the users for any irregularity in the

reading, which allows users to take precautions to make sure they are maintaining their health status, which is lacking in the "An IoT-Based Health Monitoring System for Elderly Patients" research paper. The need for alert message is crucial as the system focuses on elderly patients who are more vulnerable to health issues.

"Development of Blood Oxygen Level, Heart Rate And Temperature Monitoring System by Using ESP32" system developed by (Ahmad, et al., 2022) has presented their system using ESP32 integrated with MAX30102 and MLX90614 for SpO_2, Beat Per Minute (BPM) and body temperature respectively, which will then be displayed on the android application. The ESP used in the development of the system is powered by using LIPO rechargeable battery via charger module. ESP 32 WROVER B used in the Smart Vital project has a built-in battery holder which can power the microcontroller without the need for extra components or modules. The results can be monitored on the Serial Monitor of Arduino IDE and via cloud system. One of the biggest drawbacks of the system I think is not using a remote tool for displaying data, additionally not including mobile application. Smart Vital project contains both OLED display and mobile application for user convenience. The accuracy rate of 95% for temperature, can be improved by using sensor such as per (Sari, et al., 2021), an accuracy of 99% was achieved with right calibration. Similarly,10K NTC thermistor thermometer sensor was used in Smart Vitals which gives more accurate reading when calibration is done right. (Ahmad, et al., 2022) has stated in their research paper on how the system can be further improved, by adding blood pressure measurements, which is implemented in Smart Vitals project.

(Sheikh , et al., 2024) has delivered a simple system in "Analysis of Patient Health Using Arduino and Monitoring System" for measuring vital signs such as temperature, heart rate and blood oxygen level, processing data via Arduino. While making any device which includes crucial data it is important for the developers to choose the components wisely specially health monitoring system.

The study carried out by (Hosan, et al., 2025) concluded that ESP is reliable as it performs much better in comparison to Arduino while performing sophisticated tasks. “Analysis of Patient Health Using Arduino and Monitoring System” does not focus on development of mobile application, only the use of LCD is seen. Smart Vitals has mobile application which allows users to explore different features. (Sheikh , et al., 2024) has stated in their journal about inaccurate readings and less IoT features. It is essential to have nearly precise reading while developing such system so that users can have assurance and can take precautions accordingly.

CHAPTER 3: REVIEW OF TECHNOLOGY

Different tools, technology and components were used in the project during the development of Smart Vitals which are listed below:

3.1 Hardware Components

1. **ESP 32 WROVER TYPE B** is a micro-controller; it has powerful ability to process data with built-in Wi-Fi and battery holder with many GPIO pins which was the reason for its usage.

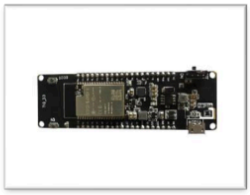


Figure 1: ESP 32 WROVER TYPE B

2. **MAX30102 sensor** uses photoplethysmography (PPG) which was used to measure the heart rate, SpO2 and an estimated blood pressure.

Figure 2: MAX30102 sensor

3. **AD8232** is a device to measure ECG signal. It was used as it gives optimal ECG signal measurement.

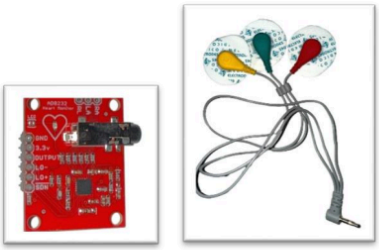


Figure 3: AD8232 sensor

4. **10K NTC Thermistor temperature sensor** is used for measuring body temperature. Its resistance lowers while temperature is increased, making it suitable for accurate temperature measurement.

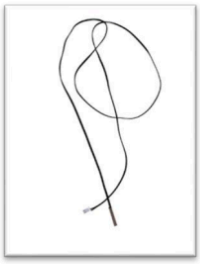


Figure 4: 10K NTC Thermistor temperature sensor

5. **10 K Resistor** is electrical component which controls voltage and limit current. It ensured the components receive proper voltage and current supply. It was used in the 10 K NTC thermistor thermometer sensor.

Figure 5: 10 K Resistor

6. **Push Button** is electromechanical switch, allowing manual control of the circuit. It was implemented to switch the sensors measurement as per requirement.

Figure 6: Push Button

7. **OLED** was used for local display of the real-time measurements also displaying necessary messages.

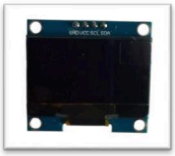


Figure 7: OLED display (1.3' inch)

8. **Batteries** were used to provide a common power supply to make sure the micro controller worked properly with all the sensors.

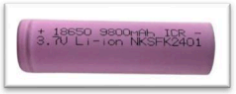


Figure 8: Li – ion Battery

9. **MQ-135** was used for air quality monitoring of a particular room, showing different ranges of air quality.

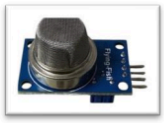


Figure 9: MQ-135 sensor

10. **NEO-6M GPS Sensor** was used as it allows users to track the location, which is used as additional layer for safety of an individual to take necessary precautions.

Figure 10: NEO-6M GPS Sensor

11. **DHT22** sensor was used to as it gave the insights on the current room temperature.

Figure 11: DHT22 sensor

12. **Matrix Board** was used to integrate all the hardware components with ESP 32 Wrover type B making the connection easier to join and compact.

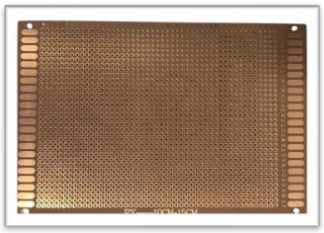


Figure 12: Matrix Board

13. **Soldering items** such as soldering rod and soldering wire, steel wire wool, wax was used to join the wire for fixed connections.

Figure 13: Soldering iron, wax, wire, steel wire wool

14. **Glue** was used to join all the components ensuring the connections to work properly and to attach the 3D printed box.

Figure 14: Hot Glue Gun

15. **Connection Wire** was used alongside soldering to connect the micro controller with the hardware components.

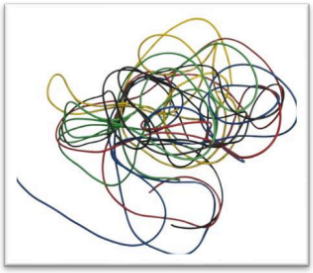


Figure 15: Connecting wire

16. **3D printing (Creality Ender 3)** was used to make a stable box which fitted all the hardware components which made the project more presentable and easier to carry.

Figure 16: Creality Ender 3

17. **Power bank Module** was used to charge the battery via charging cable also to power the system (micro controller and all the hardware components).

Figure 17: Power bank module

18. **Reset Button** was used as it allows to reset the ESP, which might come in handy when the system is lagging or when OLED display is not responsive.

Figure 18: Reset Button

19. **Type B cable** was used to connect the system with power bank module. It was also used to charge the module.

Figure 19: Type B cable

3.2 Backend

1. **Firebase** was used as it acts as a secure platform for user authentication and stores users' current health vital information ensuring smooth access to user data. It allows to create collections which was further divided into documents to store user data.

3.3 Operating System

1. **Windows 11** is used as it supports all the software and hardware components for smooth operation of the SmartVitals project.

3.4 Language, framework and Libraries

1. **C/C++** was used as primary programing language as it helps in effective programing and suitable for hardware like ESP32.
2. **Dart** was adopted for development of mobile application.
3. **Flutter** was used for creating responsive UI for real-time data visualization.
4. **ThingSpeak** allowed to visualize, collect and analyze data after integration with microcontroller.
5. **ESP32-SDK** allowed effective development, handle wireless communication and hardware integration.
6. **Wire.h** was used for I2C communication for effective data exchange among ESP32 and components used.
7. **WiFi.h** library was used as it allows Wi-Fi connectivity for IoT devices which helps in data transmission.
8. **SparkFun MAX3010x** Library allowed easy communication with MAX sensor and collect accurate data in real-time.
9. **U8g2** library is common library for graphics display used with microcontrollers to get display in OLED.
10. **Math.h** library was used to carry out mathematical calculations which was essential during calibration and while using different formulas and equations.
11. **TinyGPS++.h** library was used to gather GPS data which helped in getting the location of the device.
12. **DHT.h** library was utilized while gather data from DHT22 sensor for temperature data.

13. **HTTPClient.h** library was used to receive and send data to and from the web server.
14. **HardwareSerial.**h library was used as it allows communication with various serial device using UART ports, which are built-in ports in ESP32.

3.5 Version Control

GitHub was utilized as version control as it helped to manage the codes and files efficiently and keep track of the project history and encouraged for regular commits. Project progression was also seen through commit history.

3.6 Documentation

1. **MS Word** was used to create initial plan, ethical consent form for the project and prepare a thorough report of the project.
2. **MS Project** was used to present a well-constructed workflow of the project for ease development of the project in a structured manner.
3. **MS PowerPoint** was used to deliver presentation which supports the development of the product.

3.7 Web Browser

Google Chrome was utilized as it is easy to use, offers secure surfing experience and is trusted web browser for debugging and running web portals.

CHAPTER 4: METHODOLOGY

Agile methodology has been used for Smart Vitals project. Agile method allows scalability and adaptability which enables to handle technical error effortlessly at early stages without interrupting other components. The flexibility, in Agile methodology allows developers to include new features or components in later stages based on the evolving requirements. Constant feedback is provided which ensures the progression of the project to meet the end goal the expectations. Additionally, the agile method offers iterative process enabling to build and improve the project piece by piece rather than following a linear or firm plan.

The Agile process has different stages each stage contributing for the progressive development of the Smart Vitals project. Research and Specification Collection is the first stage where the essential modules like ESP32, MAX30102, SpO_2 sensor, ECG sensor, Temperature sensor, are examined with the help of the datasheets. The next stage is Planning and Circuit Designing where the sensor connections were validated with the help of Wokwi for circuit diagram layout. After the successful circuit design, the main focus was integration of MAX30100, ECG and Temperature sensor with ESP32 for processing sensor data and with OLED for data monitoring. Then, 3D models and PCB design was developed and designed in TinkerCAD during Designing and Prototyping stage. In the Development stage, the microcontroller was programmed to analyze sensor data to communicate wirelessly with the mobile app. After the system being functional, in the Testing phase the product was regularly tested ensuring precision, stability and working of hardware and software components. Lastly, in deployment and maintenance stage, improvement of the product was done where the product was updated based on user feedback.

The waterfall method is a conventional and sequential approach where each stage must conclude before beginning with the next stage following stable workflow used for project that has well-defined end goal. The project Smart Vitals project has different hardware and software integration which needs to be tested continuously for precision and improvement, in waterfall method testing is

conducted following the development stage. Any issue in the project might not be identified until the very end which will be very risky making the process time consuming and complex which is the reason for Agile methodology being used as it offers constant testing and evaluation throughout the product development. RAD focuses on prompt development of the prototypes and user reviews. The Smart Vitals project focuses on providing accurate, dependable and secure data which are essential for health care, RAD focuses on quick deployment which could result in patchy testing and disparity between hardware and software. Therefore, Agile methodology is used as it provides iterative feedback which assures the product is tested thoroughly and reliable medically. The Spiral Model mainly focuses on risk management and developing the project in an iterative manner which can help the developers eliminate or avoid the risks completely during the initial phase. It focuses on thorough reports, planning which is suitable for large-scale project with peak risks which is why agile method is used as it is suitable for small to medium scale projects like Smart Vital with risks like data reliability which can be avoided with regular evaluation.

CHAPTER 5: PRODUCT DESIGN

5.1 General Overview of the system

Figure 20: Smart Vitals device

Smart Vitals system starts the work when the device is turn on via plugging the cable, which activates the micro controller then the OLED display. The OLED asks the user to press the button according to their preference of vital measurement which then activates the sensor, the sensors start taking the reading. The reading via sensor is seen on the OLED display. The user can press the button again to deactivate the reading. Additionally, reset button has been placed which allows the users to reset the system when the system does not response. For charging the battery, Type B cable can be used.

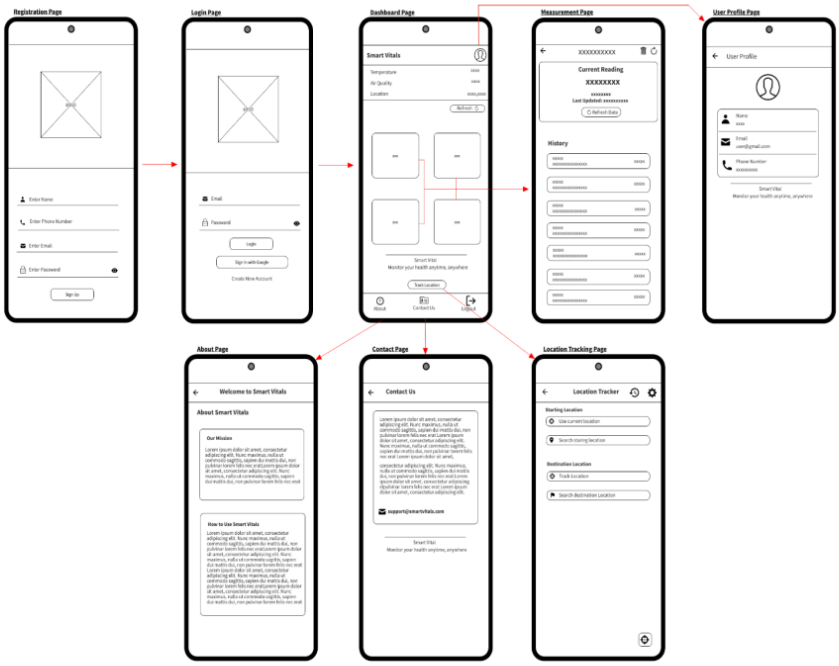


Figure 21: Wireframe for Smart Vitals

When the user logins on the Smart Vital mobile application, they can get access to their desired reading. If the user is not already registered the user can create a new account with valid credentials to get access to their vital readings. Additionally, the user can also view their vital history. The user can also get access to their profile, get general overview of the Smart Vital, how to use it on about page. The contact us page is also seen on the interface which allows user to get in touch if any queries. The user gets alert like voice alert and mail alerts if there are any abnormalities in the reading.

5.2 Data Flow Diagram

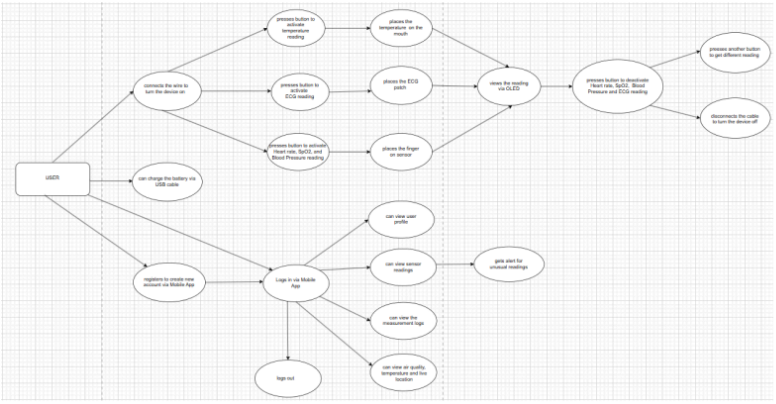


Figure 22: Dataflow diagram for Smart Vitals

Smart Vitals lets users to view their health vitals remotely and via mobile application which allows them to stay on top of their health and take necessary precautions. The smart vital is designed in such a way that the users can easily operate the device. To take any measurement the user first needs to turn on the device via button, which is present on the side. Once the device is turned the user can see the OLED being turned on, asking them for the specific vital they want to measure. The user needs to press the button according to their need of measurement, SpO2, Heart rate, Blood Pressure or ECG or body temperature. The system starts taking the reading once user has placed their finger on the sensor for SpO2, heart rate and Blood Pressure reading, or placed the patches on for ECG reading or placed the thermometer for temperature reading. The user can view their health readings on the OLED for remote display. Once the reading is done, they can either deactivate the sensor by pressing the button or turn the device off. The user can also charge the battery via USB cable.

Additionally, the user can also register to the mobile application of smart vitals with their credentials or login if they already have an account with their correct

credentials, which will redirect them to the smart vitals' application. The user can view their health vitals via mobile application; all the measurements are separated so that it is easier for user to view the desired readings. They can also view their profile, can check the health measurement history. User can also view about smart vitals page to get information related to device. The user can get insights on the air quality, environment temperature and live location. The user can log out of the system with ease at any moment.

CHAPTER 6: SOFTWARE REQUIREMENT ANALYSIS

The table below represents the features of the project with its priority listed as HIGH, MEDUIM OR LOW. HIGH being the most important feature, followed by MEDIUM being moderately important and LOW being relatively less important feature of the product.

Features	**Priority**
The device should be easy to use and portable.	HIGH
Cable for turning the device on and off.	HIGH
The device should connect to stable Wi-Fi connection for flawless communication.	HIGH
Push buttons for switching the readings between the sensors for particular health measurement as per user's desire.	HIGH
Readings from sensors such as MAX30102, 10K NTC Thermistor Thermometer sensor, AD8232.	HIGH
The reading taken via sensor should be displayed on the OLED with appropriate message.	HIGH
The measurements should be shown in the mobile application after bring connected to the stable Wi-Fi connection.	HIGH

Voice Alert should be sent to the user if there is any abnormality in the reading.	MEDIUM
Additionally, alert via email is also sent to the user when any abnormal reading is detected with device location.	MEDIUM
The data from the measurements should be logged when user desires.	MEDIUM
The device should be able to charge when connected using USB cable Type B.	MEDIUM
Users can get access to the live location from mobile application when device is turned on and connected to a stable Wi-Fi connection.	MEDIUM
Device location coordinates (latitude and longitude) should be seen on the home page when exposed to open sky.	MEDIUM
User can delete the vitals history whenever they want.	LOW
User can delete their profile as per their desire.	LOW
Additionally, user can view location history.	LOW
Users can get insights on room temperature.	LOW
Air quality monitoring should be seen on the homepage of the mobile application.	LOW
Reset Button should refresh or reload the OLED display.	LOW

Table 1: Software Requirement Analysis for Smart vitals

CHAPTER 7: IMPLEMENTATION AND TESTING

7.1 Setting Device

7.1.1 Arduino IDE Board Manager

The esp32 board by Espressif Systems is used as it supports ESP 32 Wrover Module which is the ESP used in the development of the smartvitals project.

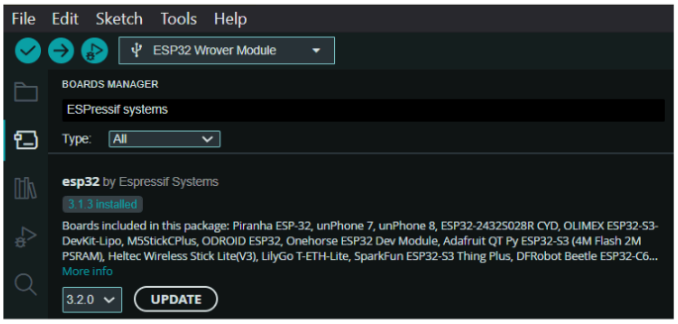


Figure 23: esp32 Board Manager

7.1.2 Arduino IDE Libraries

Different libraries were used for the hardware components for the successful project completion. All the libraries used are listed below:

1. Spark Fun MAX3010X library speeds up the development process and reduces bugs with its predefined functions, which is used to calculate the

pulse rate, and SpO2.

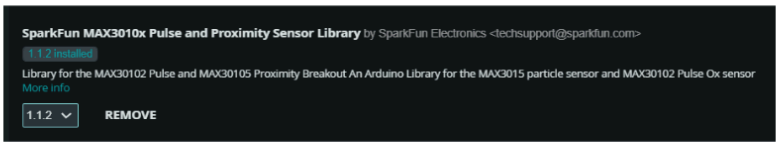


Figure 24: SparkFun Max3010x Pulse and Proximity Sensor Library

2. WiFi Manager library provides a simplified way to get connected with the wifi, creating a portal allowing easy connection when a new network is being used.

Figure 25: WiFi Manager Library

3. DHT sensor library provides a ready to use function such as readTemperature() which does the signal reading and timing in the background and gives the value for the temperature.

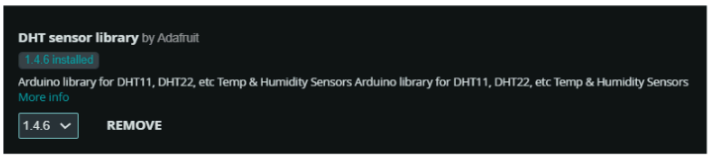


Figure 26: DHT sensor Library

4. TinyGPSPlus library gives a parsed, readable value for longitude and latitude from the raw data coming from the Serial of the GPS without the need for manual decoding.

Figure 27: TinyGPSPlus Library

5. U8g2 library allows to easy communicate with the OLED, with I2C interfaces with monochrome display.

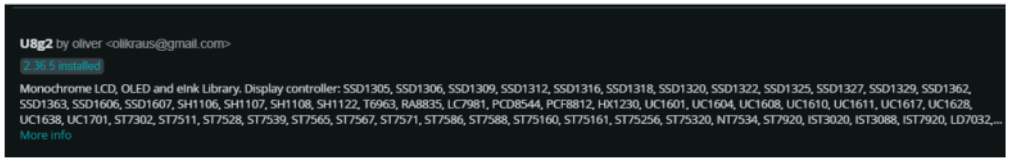


Figure 28: U8g2 Library

7.1.2 Pin Configuration

The screenshots below show the pins assigned to each hardware components. For OLED and MAX30102 pins 5 and 18 for SDA and SCL are assigned respectively. Pin 26 for RX and pin 27 for TX is assigned for GPS module.

```
136   Wire.begin(5, 18); // SDA, SCL for OLED/MAX30102
137
138   gpsSerial.begin(9600, SERIAL_8N1, 26, 27);  // GPS Module: Baud, Format, RX, TX
139
```

For ECG output, pin 33 is assigned, for thermistor pin 34 is assigned similarly pin 4 and pin 35 is assigned for analog read for DHT22 and MQ-135 sensors respectively.

```
// ECG
#define ECG_PIN 33

// Thermistor configuration
#define SENSOR_PIN 34

// DHT sensor for temperature
#define DHT_PIN 4

// MQ-135 sensor for air quality
#define MQ135_PIN 35
```

Three pins 12, 14 and 15 are assigned for buttons to operate MAX30102, ECG and Thermistor sensor correspondingly, which when clicked should activates the sensors.

```
// Button pins
#define BUTTON1_PIN 12  // MAX30102
#define BUTTON2_PIN 14  // ECG
#define BUTTON3_PIN 15  // Thermistor
```

7.1.3 WIFI Configuration

The code below starts a WIFI portal with “Smart Vitals” as its name which allows users to set up the network. Then it confirms if micro controller is connected before transmitting to the API.

```
WiFiManager wifiManager;
Serial.println("Starting Wi-Fi Manager...");
wifiManager.autoConnect("Smart Vitals");
```

```
    if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(url);
    int httpCode = http.GET();

    http.end();
  } else {
    Serial.println("WiFi not connected.");
  }
}
```

7.1.4 OLED Display with Feedback

The code snippet below shows how the OLED displays messages according to users' actions, when the system starts the choice for pressing the buttons are seen.

```
151   u8g2.clearBuffer();
152   u8g2.setFont(u8g2_font_ncenB08_tr);
153   u8g2.drawStr(3, 12, "Smart Vitals Ready");
154   u8g2.drawStr(3, 28, "Btn1: HR/SpO2/BP");
155   u8g2.drawStr(3, 44, "Btn2: ECG");
156   u8g2.drawStr(3, 60, "Btn3: Temperature");
157   u8g2.sendBuffer();
158 }
```

When button one is pressed once it displays, "MAX30102 Activated!", when button one is pressed again, "MAX30102 Deactivated" is seen, which ensures accessibility to the users.

```
198   u8g2.clearBuffer();
199 u8g2.setFont(u8g2_font_ncenB08_tr);
200 u8g2.drawStr(3, 12, "Smart Vitals");
201 u8g2.drawStr(3, 28, maxSensorActive ? "MAX30102 Activated!" : "MAX30102 Deactivated!");
202
203 u8g2.sendBuffer();
```

If the IR value is less than 50000, the message below is seen similarly, if no finger is detected on the sensor, the message is displayed.

```
307   if (irValue < 50000) {
308     Serial.println("Please place your finger on the sensor.");
309     u8g2.clearBuffer();
310   u8g2.setFont(u8g2_font_ncenB08_tr);
311   u8g2.drawStr(3, 12, "Smart Vitals ");
312   u8g2.drawStr(3, 28, "Please place your");
313   u8g2.drawStr(3, 44, "finger on the sensor.");
314   u8g2.sendBuffer();
315     delay(1000);
316     return;
317   }
```

7.1.5 Hardware Enclosure and Final Assembly

After the system was ready, it was necessary to make the system compact and the connection to be intact. So, all the hardware components were placed inside a custom printed 3D enclosure using Creality Ender 3, a 3D printing machine. Additionally, a separate container was also printed for power bank module and li – ion batteries.

Figure 29: Hardware Enclosure of Smart Vitals

7.1.6 Dependencies in Flutter

The “pubspec.yaml” file below consist of all the dependencies like -maps, tts, geolocator, mailer, flutter_dotenv, etc. including all the settings which are required to run and build the Flutter based mobile application for the project, SmartVitals.

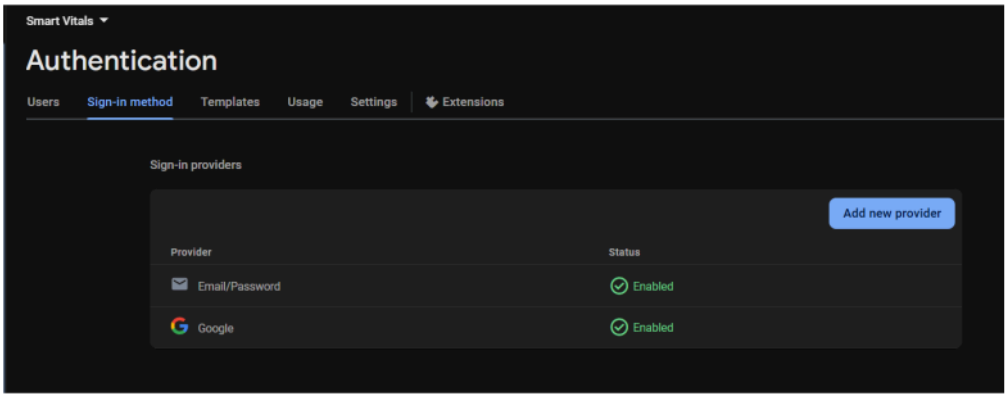
```yaml
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  http: ^1.3.0
  fl_chart: ^0.71.0
  font_awesome_flutter: ^10.8.0
  firebase_core: ^3.13.0
  firebase_auth: ^5.5.2
  google_sign_in: ^6.3.0
  provider: ^6.1.4
  cloud_firestore: ^5.6.6
  firebase_messaging: ^15.2.5
  firebase_storage: ^12.4.5
  get: ^4.7.2
  shared_preferences: ^2.5.3
  intl: ^0.20.2
  path_provider: ^2.1.5
  url_launcher: ^6.3.1
  geocoding: ^3.0.0
  latlong2: ^0.9.1
  geolocator: ^14.0.0
  flutter_map: 6.0.0
  sqflite: ^2.4.2
  flutter_tts: ^4.2.2
  mailer: ^6.4.1
  flutter_dotenv: ^5.2.1

dev_dependencies:
  flutter_test:
    sdk: flutter

```

7.1.7 Firebase Configuration in Mobile Application

The google-service.json file code snippet is given below, the file consists of configuration details required for connecting the firebase services like, database, authentication and storage with the flutter mobile application. It includes configuration details like: project number, project id, firebase services, and Api keys.

7.1.8 Sign – in method in Firebase Authentication

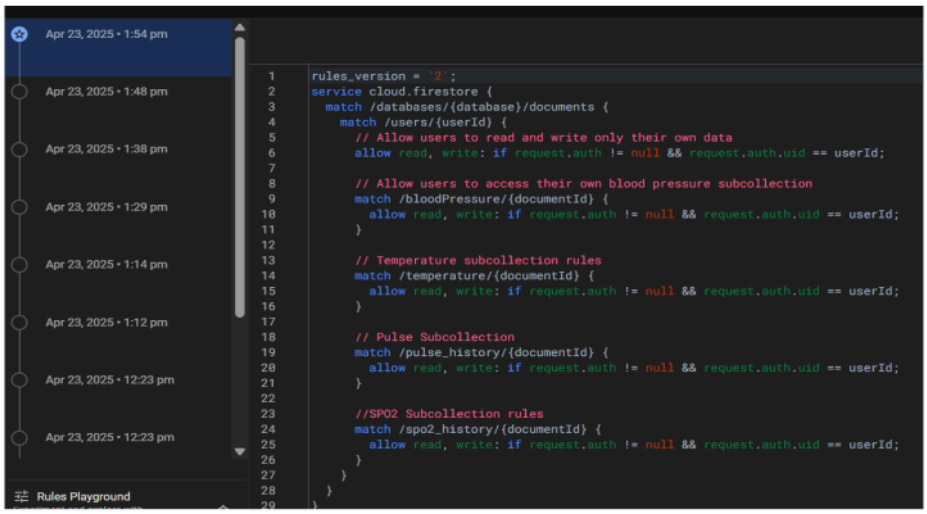
The screen shot below shows the method through which the users can sign in to the smart vitals app. There are two sign – in providers Email/Password and via Google account.



7.2 Key Code Components and Logic

7.2.1 Firebase rules for read and write

The screenshot shows the read and write access given to authorized users, the user can only access those files that are self-uploaded. The conditions are checked before allowing any user to access the data or update their data.



7.2.2 Account Creation via Firebase

The code below shows, the creation of new user account with email and password in the Firebase Authentication. Then updated the account created with the user's name, and then retrieving the unique UID which is used to link with the firestore. The user data is stored in users' collection in Firebase with document for user information.

```
// Create user with Firebase Authentication
FirebaseAuth.instance
    .createUserWithEmailAndPassword(
      email: emailController.text.trim(),
      password: passwordController.text,

    )
    .then((userCredential) async {
      // After creating the user, update the user's profile with the name
      await userCredential.user?.updateProfile(
        displayName: name.text,
      );

      // Get the user ID from Auth
      String uid = userCredential.user!.uid;

      // Create a document in Firestore's users collection
      await FirebaseFirestore.instance
          .collection('users')
          .doc(uid)
          .set({
            'name': name.text,
            'phone': phoneController.text.trim(), // Added phone number
            'email': emailController.text.trim(),
            'uid': uid,
            'createdAt':
                FieldValue.serverTimestamp(),
            'lastLogin':
                FieldValue.serverTimestamp(),
            // Add any additional user data you want to store
          });
```

7.2.3 User Authentication for Login

When user tries to login with their email and password, the FirebaseAuth.instance check for the same user credentials in Firebase Authentication, if found the user is redirected to the homepage.

```
try {
  if (isLogin) {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: emailController.text.trim(),
      password: passwordController.text,
    );
    Dialogs.showSnackbar(context, 'Welcome back!');
```

```
if (FirebaseAuth.instance.currentUser != null) {
  Navigator.pushReplacement(
    context,
    MaterialPageRoute(builder: (_) => HomePage()),
  );
}
```

7.2.4 GPS Location for tracking device

The code snippet below fetches the current location of the device by using “getCurrentPosition( )” function. The fetched location is then parsed into user readable location, updating the UI and the text box as well, Additionally, the distance is also calculated and the location is finally saved in the location history.

```
Future<void> _getCurrentPosition() async {
  if (!await _handleLocationPermission()) return;

  try {
    setState(() => _isLoading = true);

    Position position = await Geolocator.getCurrentPosition();

    setState(() {
      _currentLatitude = position.latitude;
      _currentLongitude = position.longitude;
    });

    List<Placemark> placemarks =
        await placemarkFromCoordinates(position.latitude, position.longitude);

    if (placemarks.isNotEmpty) {
      Placemark place = placemarks[0];
      final address = "${place.subLocality ?? ''}, ${place.locality ?? ''}, ${place.country ?? ''}".replaceAll(RegExp(r'^, '), '');

      setState(() {
        _currentAddress = address;
        _startLocationController.text = _currentAddress!;
        _startLatitude = position.latitude;
        _startLongitude = position.longitude;

        if (_destinationLatitude != null) {
          _calculateDistance();
          _updateMap();
        }
      });

      // Save this location to history
      await _saveLocationToHistory(position.latitude, position.longitude, address);
    }
```

7.2.5 Voice Alert and Email Notification

The code below firstly checks the temperature readings, if reading is not within the defined threshold, voice alert is triggered which reads the _speakAlert text, which is possible via text-to-speech functionality.

```
    if (temp >= 99 || (temp<=95 && temp>0)) {
      await _speakAlert('Warning! Critical temperature detected!');
      await sendEmailFromGmail(temp);
      return 'Critical';
    }
    return 'Normal';
  }

  final FlutterTts _tts = FlutterTts();

  Future<void> _speakAlert(String message) async {
    await _tts.setLanguage('en-US');
    await _tts.setSpeechRate(0.5);
    await _tts.setVolume(1.0);
    await _tts.speak(message);
  }
```

Additionally, an Email alert is also sent, via email that has been setup in the environment variable. Along with the alert message, location has also been sent which allows users to get knowledge of the device location. When clicked on the URL of the map it should redirect user to the location.

```
final gmailSmtp = gmail(dotenv.env['GMAIL_EMAIL']!,dotenv.env["GMAIL_PASSWORD"]!);

  //email send
sendEmailFromGmail(temp) async {
  // print(location);
  final mapUrl = "https://www.google.com/maps?q=$location";

  final message = Message()
   ..from=Address(dotenv.env["GMAIL_EMAIL"]!,'Temperature Alert')
   ..recipients.add(_currentUser?.email)
   ..subject='Alert for temperature'
   ..text="Warning! Critical temperature detected! Temperature is $temp .The current location of the user is $mapUrl";

  try{
    await send(message,gmailSmtp);
  } on MailerException catch (e){
    print('message not Sent');
  }
}
```

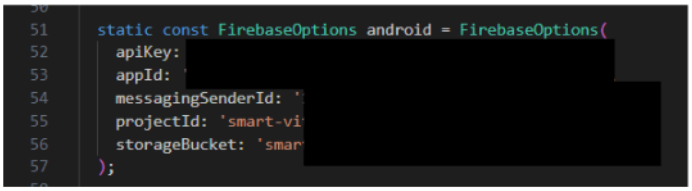
7.2.6 Storing and Retrieving data form Firestore

The vital history is saved after checking the condition, whether temperature is null or not, if it's not null then the reading is saved locally using 'SharedPreferences', then a well structure is created to store the reading to the Firestore under the users' profile.

```dart
if (newTemp != null && newTemp != 0) {
  // Save current reading to SharedPreferences for quick access
  final prefs = await SharedPreferences.getInstance();
  await prefs.setDouble('lastTemp', newTemp);

  // Create new temperature data
  final newReading = {
    'temperature': newTemp,
    'timestamp': FieldValue.serverTimestamp(),
    'serverTimestamp': serverTimestamp,
    'status': await _getTemperatureStatus(newTemp),
  };

  // Save to Firestore
  await _firestore
      .collection('users')
      .doc(_currentUser!.uid)
      .collection('temperature')
      .add(newReading);

  // Reload data to update UI
  await _fetchTemperatureHistory();
```

7.2.7 Firebase Configuration for specific platform in Flutter

The code below shows how android platform is configured in the Firebase which allows the app to initialize Firebase based on the usage of current platform.

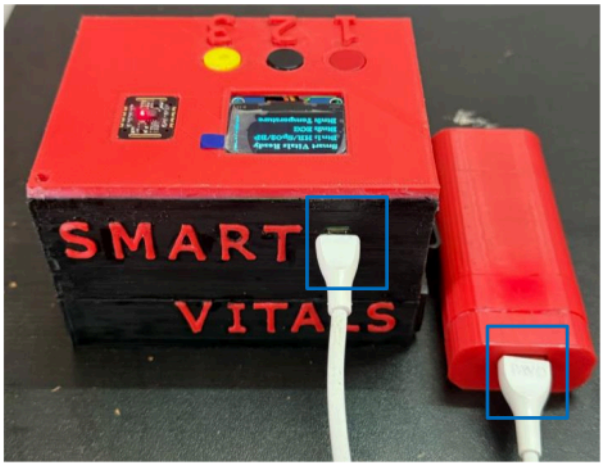


7.3 End - to - End Process Flow of the Smart Vitals

7.3.1 Device Process Flow

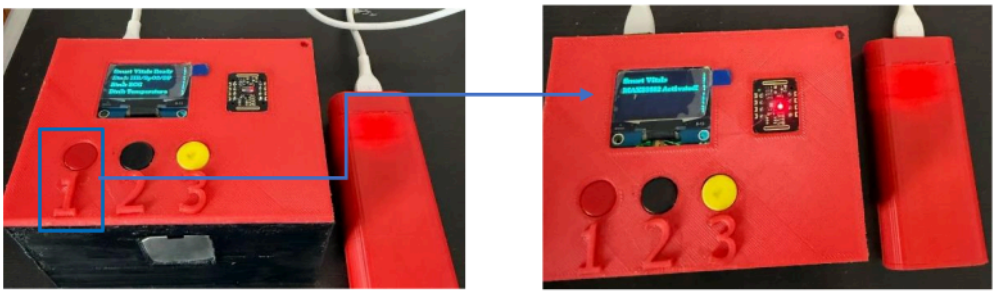
The smart vitals system is developed in such a way that it is easy to operate by the end users. The system works in a very simple way, the step-by-step guidelines on how user can access the system are listed below:

The system turns on when the user plugs the power bank module with the device, using type b cable.

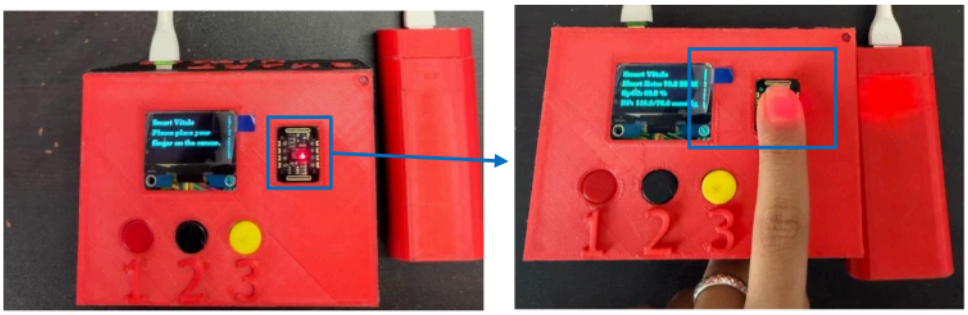


After the system is turned on, the user can view instructions on which button to press such as button1 for heart rate, SpO2 and Blood pressure, button 2 for ECG waveforms and button 3 for Body temperature measurement.

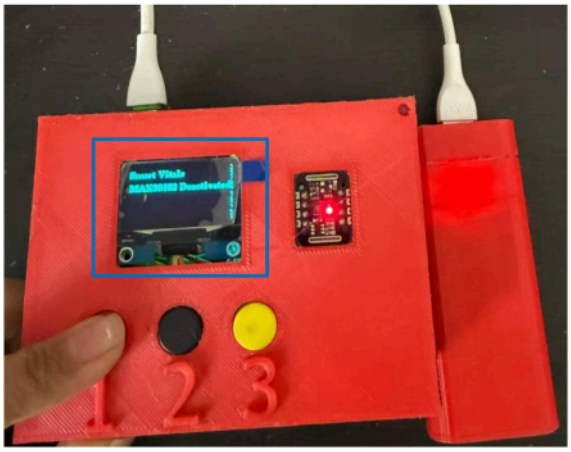
User can press button 1 to activate MAX30102 sensor which gives the readings for Heart rate, SpO2, and blood pressure.



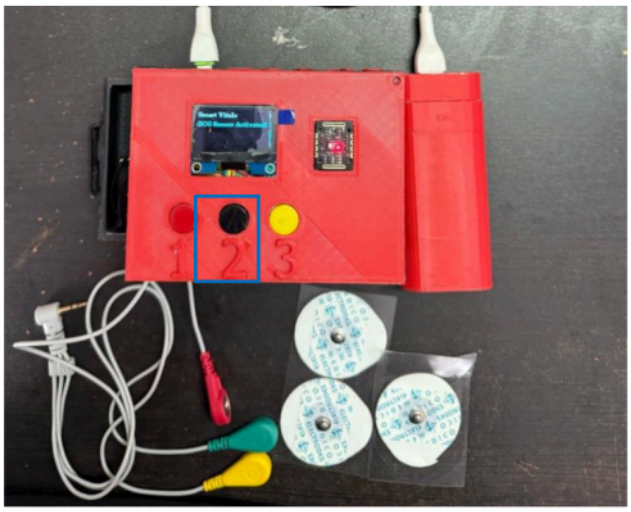
The user should place their finger on the sensor properly to take measurement which is displayed on the OLED screen placed at top of the device. When user removes their finger, the sensor stops taking the reading and displays message stating “Please place your finger on the sensor.”



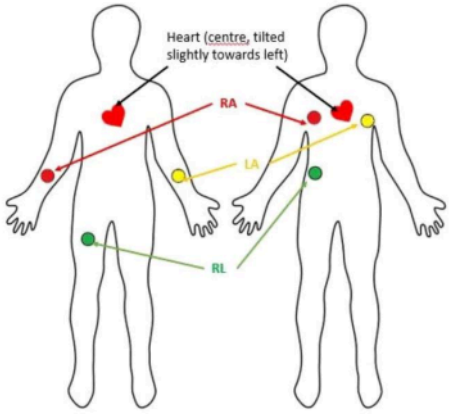
To take readings from another sensor user should press the button 1 again to deactivate the Max30102 sensor.



User can press button 2 to activate the ECG sensor. A message “ECG Sensor Activated” is seen on the OLED display. The user can then pull the drawer on the Smart Vitals box, to get access to the ECG patches.

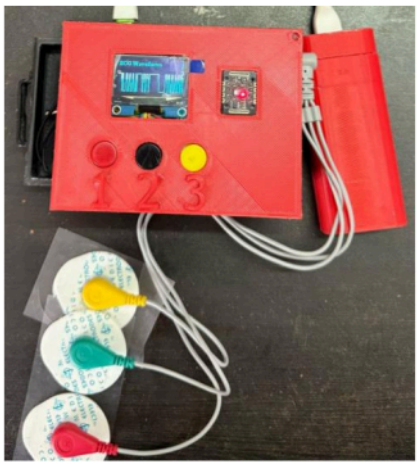


Once the ECG sensor is activated, the user should connect the cable of the patches and place the patches properly on the right arm, left arm and right leg for accurate waveform. The patches should be placed as shown in the figure below.

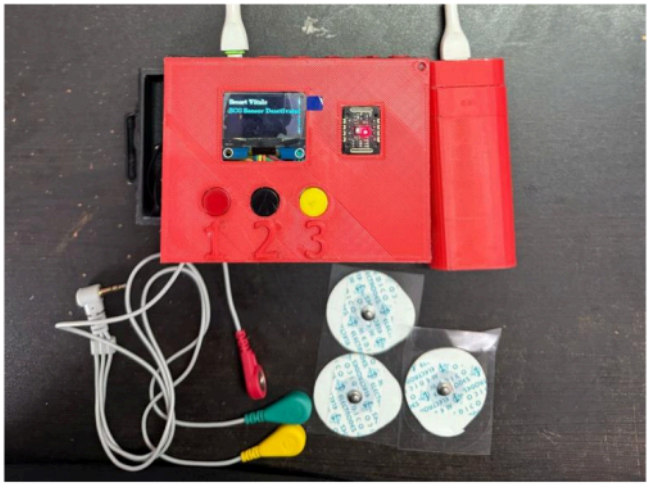


(Shah, 2021)

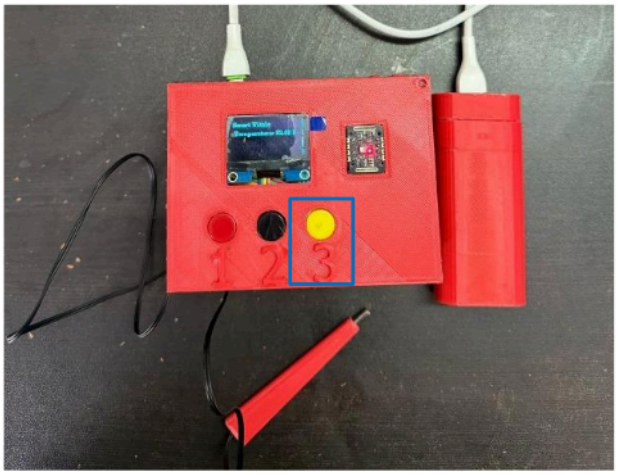
Then the waveform is displayed on the OLED screen placed at top of the device after the patches are placed correctly for proper waveform.



Similarly, to take readings from another sensor like MAX30102 and 10 K NTC thermistor thermometer, user should press the button 2 again which deactivates the ECG sensor.



User can press button 3 to activate Temperature sensor. Once the Temperature sensor is activated, the user should place the thermometer properly. It takes reading for one minute to give accurate reading. The readings from the temperature are seen on the OLED display.



User can press the reset button whenever the user faces lagging problem or to reload the OLED display, if the OLED does not respond properly.

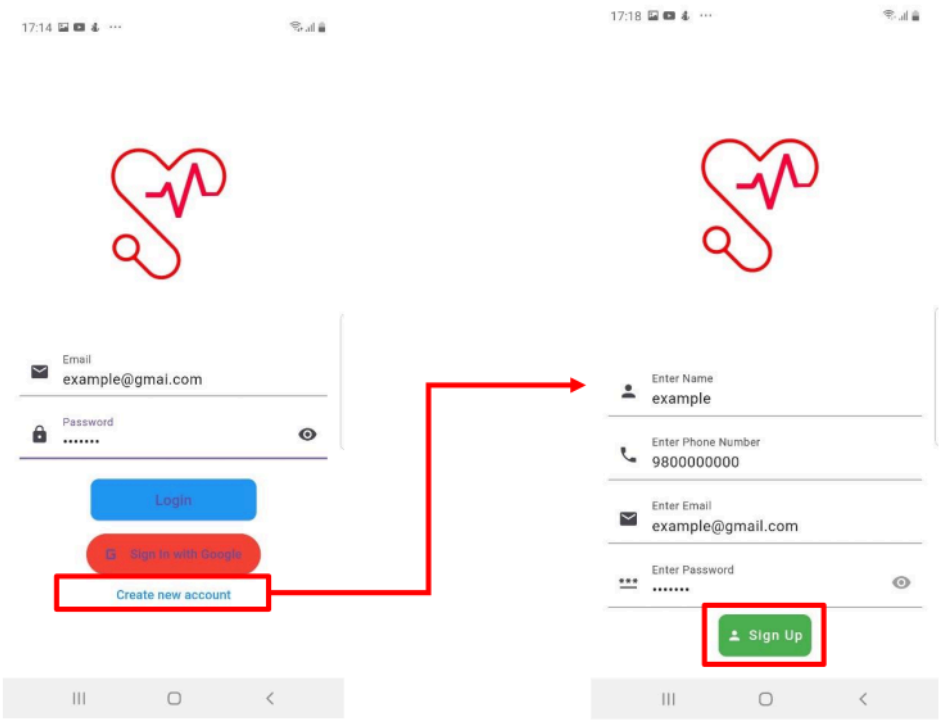
The user can charge the power bank module which they have used prior to initialize the system by connecting it with charging port.

7.3.2 Mobile Application Process Flow

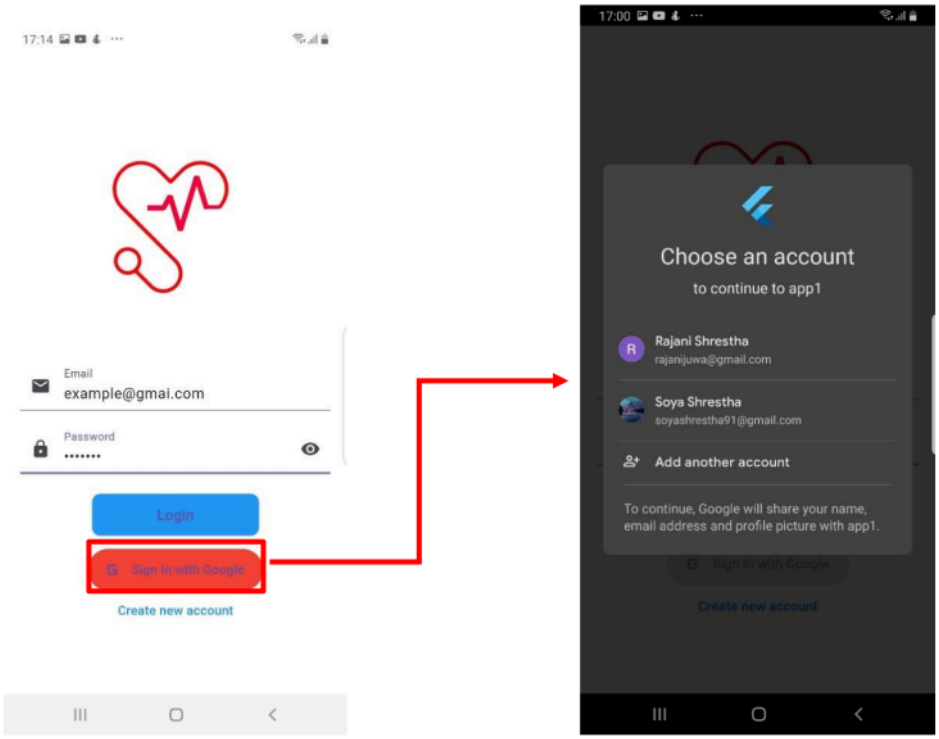
Smart Vitals system is equipped with mobile application that allows users to get benefits of many features. The step-by-step guidelines on how to get around the mobile application of smart vitals are given below:

Sign up:

User can Sign up with a valid email address, fulfilling the password criteria, their emergency contact number and user name.

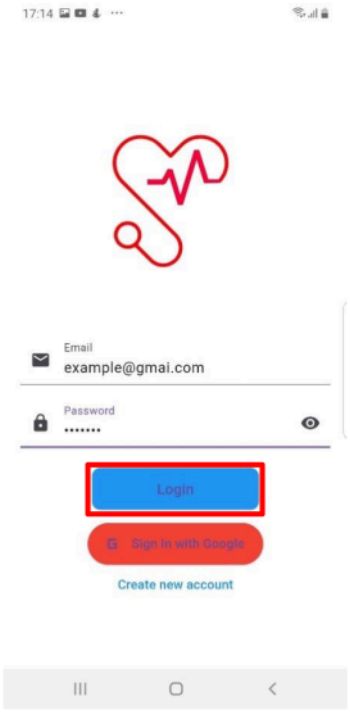


User can also sign up via Google account if they want. Once signed in from the Google account, the user is redirected to homepage.



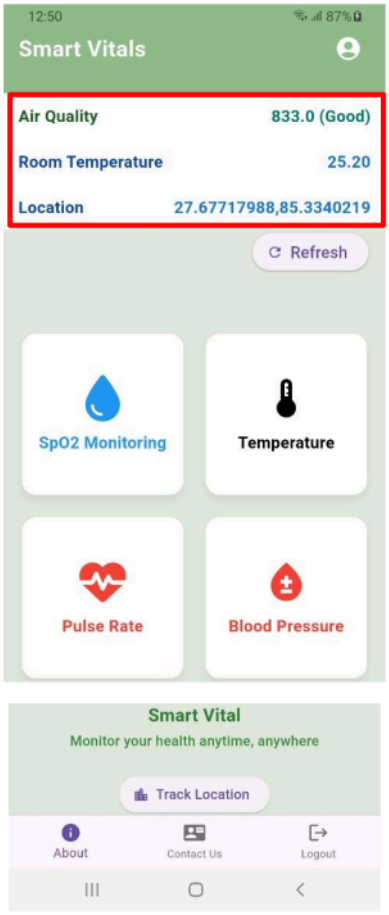
Login:

Once users' account is set up, they can login entering their previous used valid email and password and then clicking on the Login button which should redirect the user to the homepage, if there is no error.

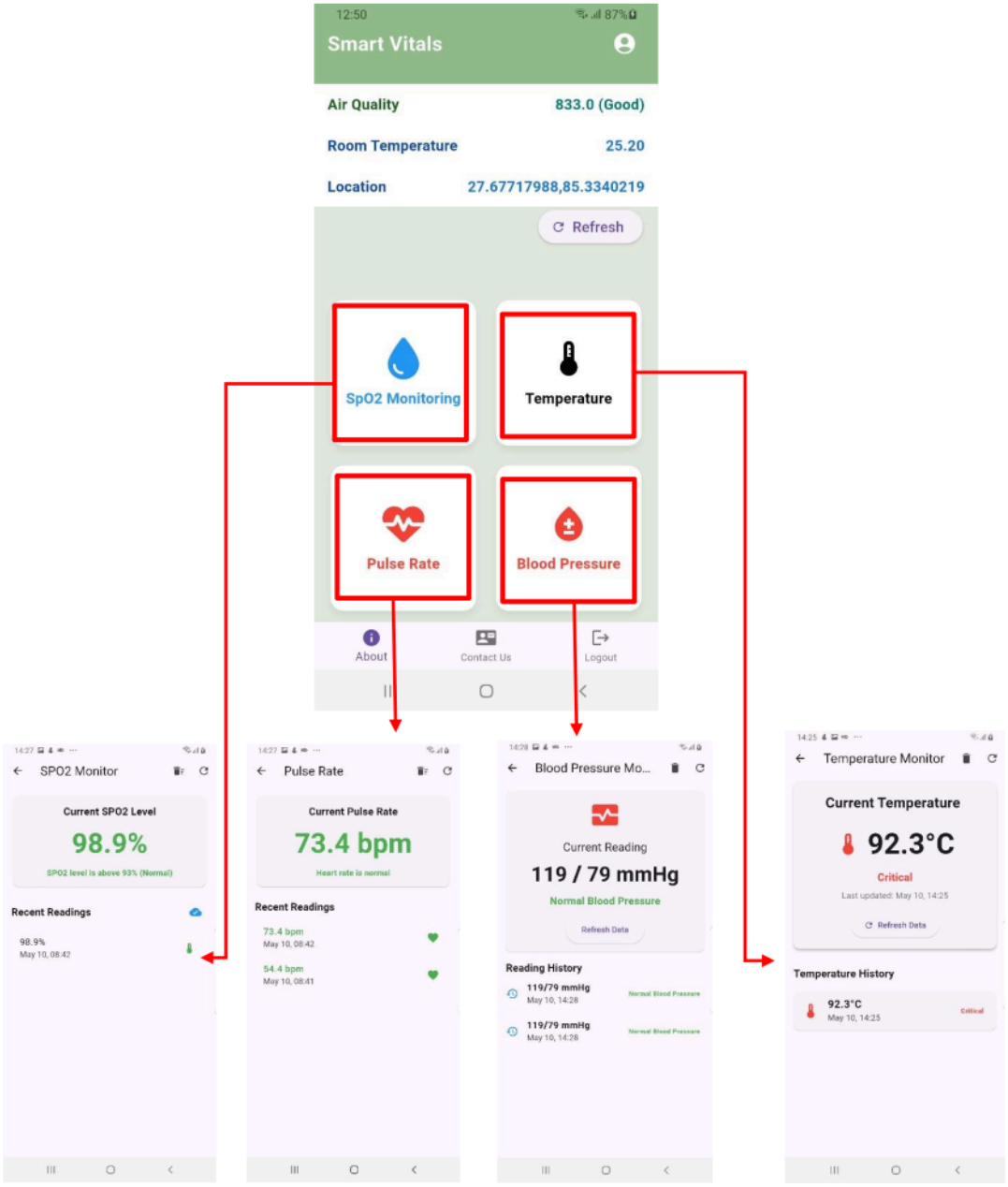


Dashboard:

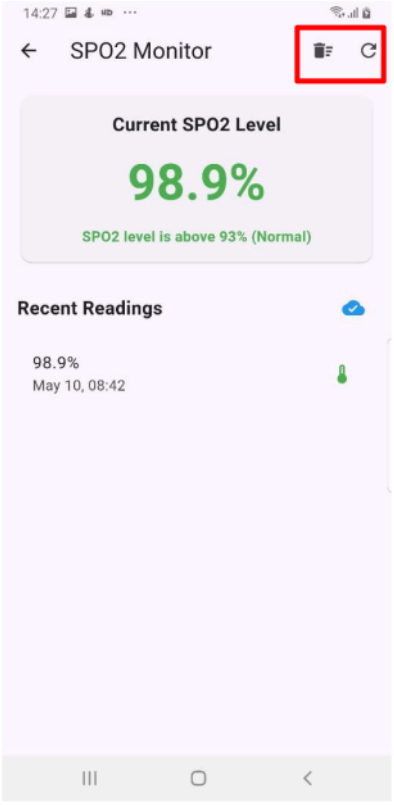
Once the user has logged in with a valid email, they are redirected to the home page. On the top of the home page, users will be able to get insights on the air quality, the current room temperature and location coordinates (latitude and longitude) of the device. User can also click on the refresh button to get the latest data on the air quality, room temperature and device location if the information is not updated.



On the main section of the dashboard users can also view, Temperature, SpO2, Heart rate and Blood Pressure cards. Clicking on any one of them redirects them to that vitals' specific page, where users can view their readings. History of vitals with date and time can be seen along with the feedback such as "Critical temperature", "Normal Blood Pressure", etc.

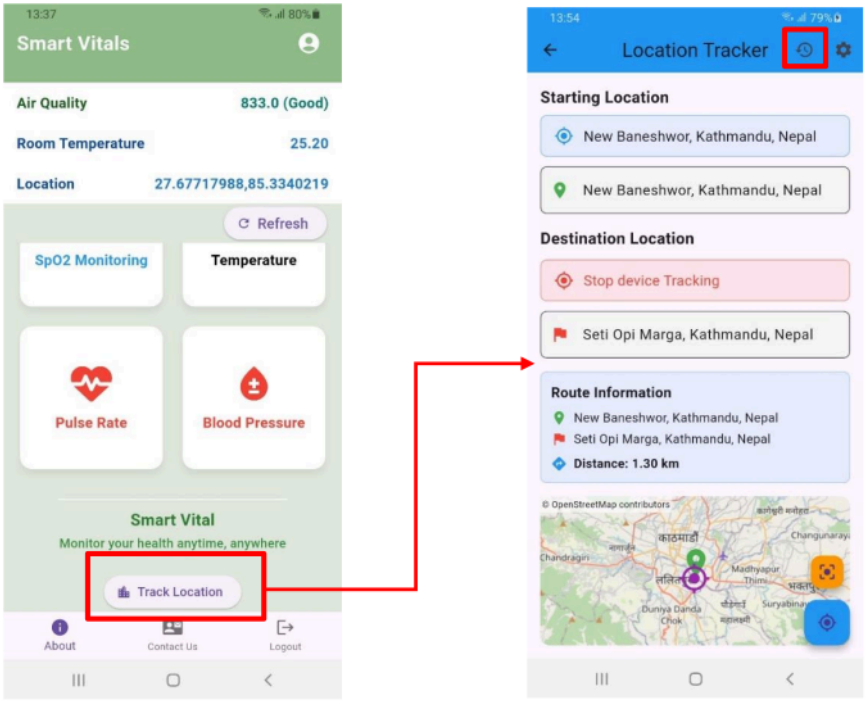

12:50
87%
Smart Vitals
Air Quality
833.0 (Good)
Room Temperature
25.20
Location
27.67717988,85.3340219
Refresh
SpO2 Monitoring
Temperature
Pulse Rate
Blood Pressure
About
Contact Us
Logout
SPO2 Monitor
Current SPO2 Level
98.9%
SPO2 level is above 93% (Normal)
Recent Readings
98.9%
May 10, 08:42
Pulse Rate
Current Pulse Rate
73.4 bpm
Heart rate is normal
Recent Readings
73.4 bpm
May 10, 08:42
54.4 bpm
May 10, 08:41
Blood Pressure Mo...
Current Reading
119 / 79 mmHg
Normal Blood Pressure
Refresh Data
Reading History
119/79 mmHg
May 10, 14:28
119/79 mmHg
May 10, 14:28
Temperature Monitor
Current Temperature
92.3°C
Critical
Last updated: May 10, 14:25
Refresh Data
Temperature History
92.3°C
May 10, 14:25

Additionally, users can also save and delete the history when they click on reload icon and trash icon respectively.



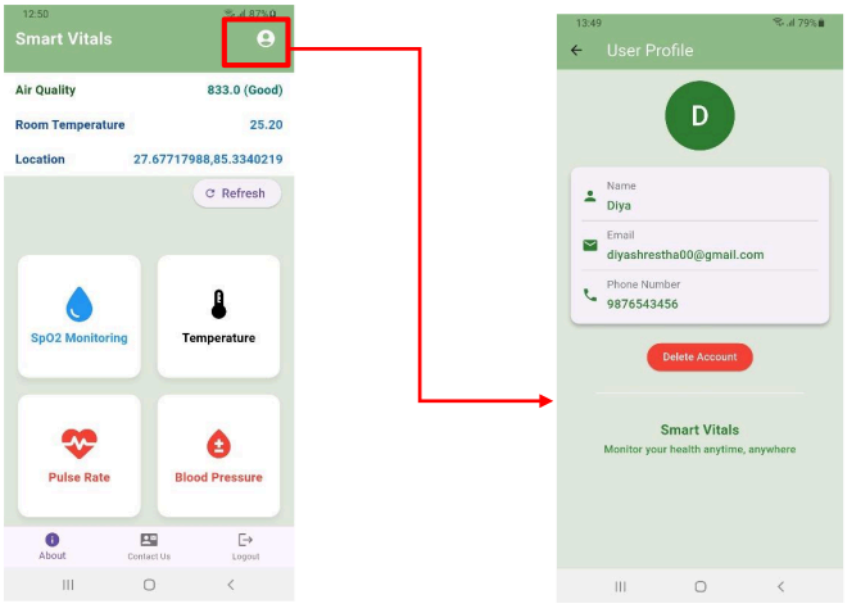
Location of the device:

When the user scrolls down, they can view track location button which redirects users to a new page where they can track the current location of the device, which comes in handy in case of emergency. Additionally, user can view location history.



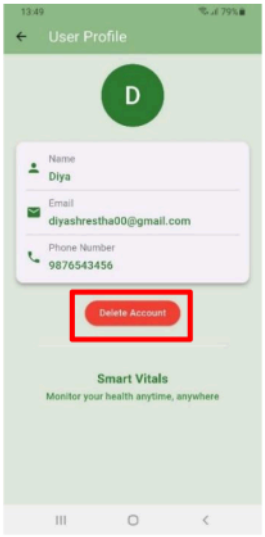
User Profile:

The user can view their information such as user name, user email, and user phone number when the user clicks on the profile icon which is visible on the top right corner of the dashboard.



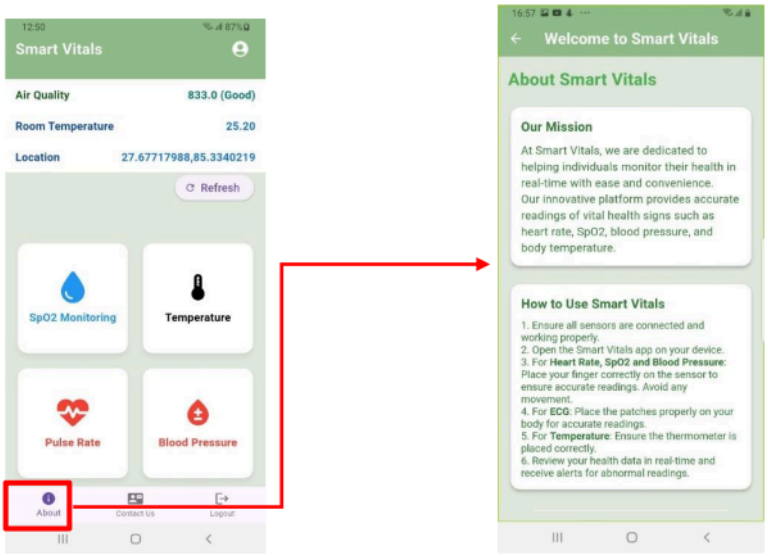
.

User can click on the delete button to delete their account from the User Profile, which deletes their account with their vitals history



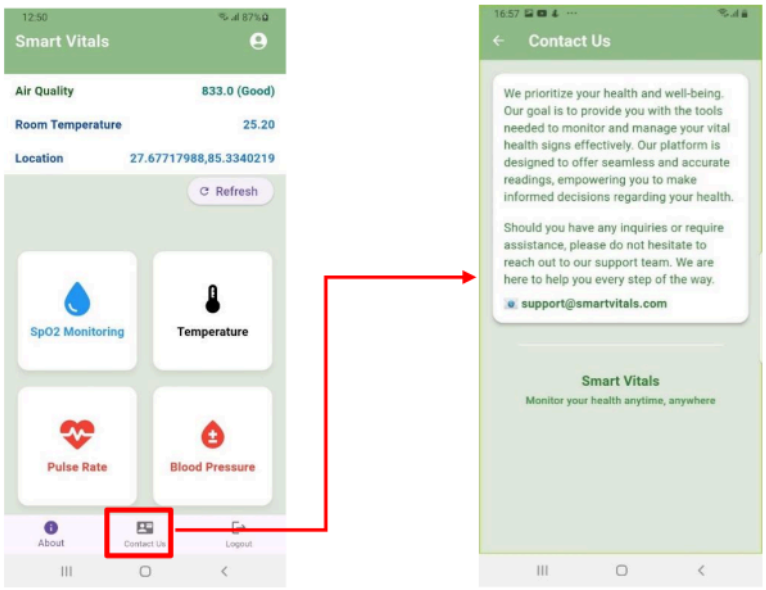
About Us:

On the bottom navigation bar, the user can see three icons, the first one is about us icon containing information about smart vitals.



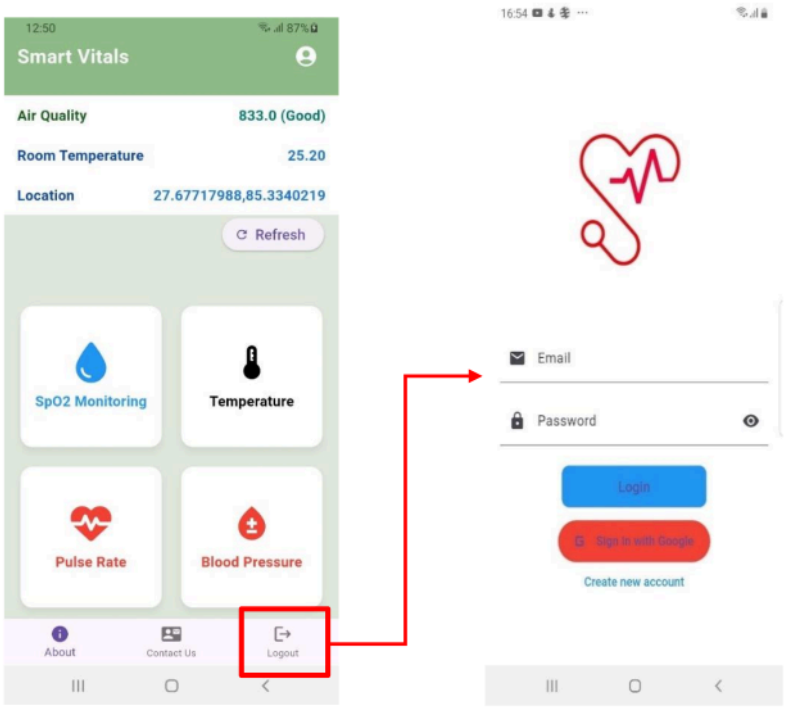
Contact Us:

The second icon is contact us, which contains contact information of smartvitals.



Log out:

The third icon is logout icon which redirect the users to the login page.


12:50
Smart Vitals
Air Quality
833.0 (Good)
Room Temperature
25.20
Location
27.67717988,85.3340219
Refresh
SpO2 Monitoring
Temperature
Pulse Rate
Blood Pressure
About
Contact Us
Logout
Email
Password
Login
Sign in with Google
Create new account

CHAPTER 8: PRODUCT EVALUATION

In this chapter, features of the system have been listed for device and for mobile application separately with their expected results, the outcome of those features and evaluation to know whether the feature is performing as stated or not.

8.1 For the Device

No.	Features	Expected	Outcome	Evaluation
1	Connect the cable	The micro controller should turn on and OLED start the display.	The micro controller and OLED is turned on.	**Pass**
2	Press the button for heart rate, SpO2 and Blood Pressure	OLED should show MAX30102 is activated.	OLED displayed “MAX 30102 Sensor Activated”.	**Pass**
3	Place the finger on the sensor	The device should start taking the reading and display it on the OLED screen.	OLED started displaying the readings.	**Pass**
4	Remove the finger from the sensor	Should ask the user to place the finger on the sensor.	“Please place your finger on the sensor” message is displayed.	**Pass**
5	Press the button for heart rate, SpO2 and Blood Pressure again	OLED should show MAX30102 is deactivated.	OLED displayed “MAX 30102 Sensor Deactivated”.	**Pass**

6	Press the button for Temperature	The device should start taking measurement for body temperature, the reading should be visible via OLED.	The device stated taking the temperature measurements and displayed on the OLED.	**Pass**
7	Press the button for ECG	The device should activate ECG sensor. “ECG Activated” message should be displayed on the OLED screen.	OLED displayed “ECG Activated”.	**Pass**
8	Place the patches	The ECG waveform should be visible via OLED.	OLED displayed ECG waveforms.	**Pass**
9	Press the button for ECG again	The device should deactivate ECG sensor showing ECG Deactivated message on the OLED display.	OLED displayed “ECG Deactivated”.	**Pass**
10	Connect Type B cable with the power bank module	It should charge the battery that powers the device.	Battery is charged.	**Pass**
11	Press the reset button	It should restart the system.	System restarts.	**Pass**
12	Disconnect the cable	The device should be turned off.	The device turned off.	**Pass**

Table 2: Product Evaluation for Smart Vitals Device (System)

8.2 For the Mobile Application

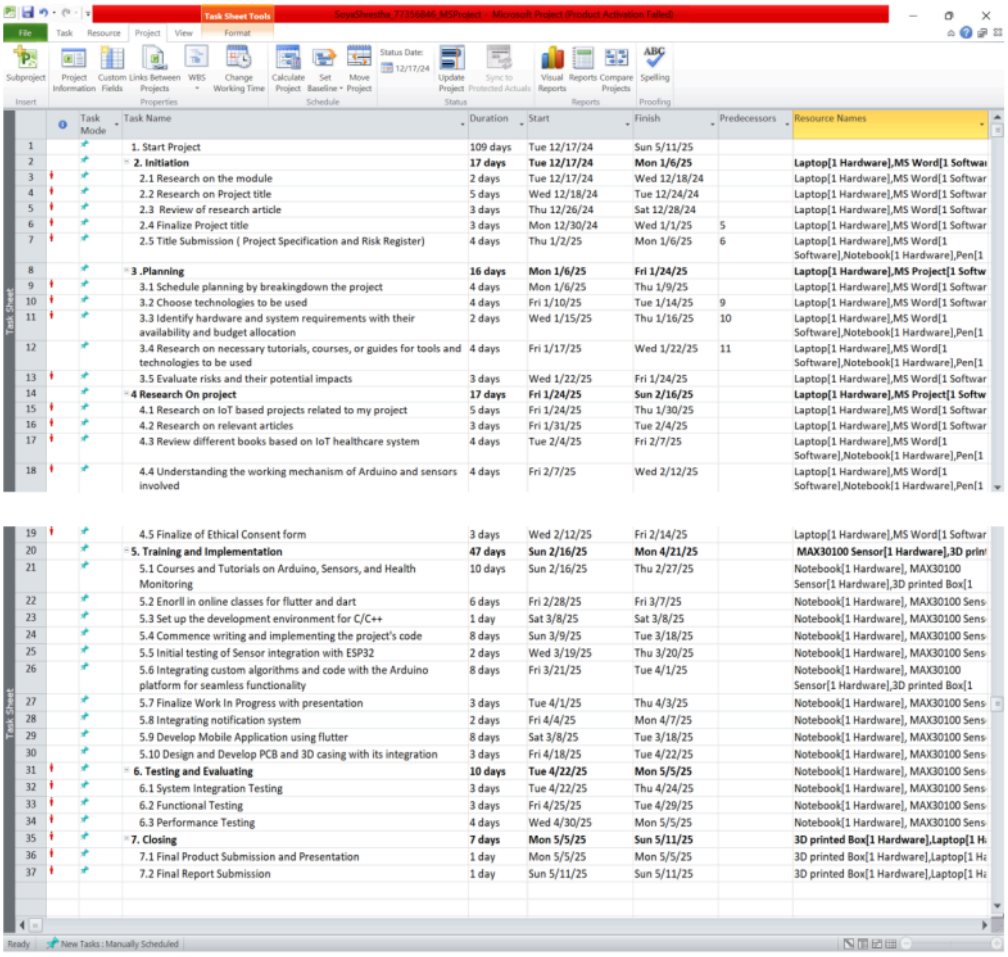
No.	Features	Expected	Outcome	Evaluation
1	User Register	User should be able to create a new account.	New account was created.	**Pass**
2	User Sign up via Google account	User should be able to create new account via Google.	New account was clicked with sign up with google account was selected.	**Pass**
2	User Login	The app should be redirected to dashboard page when logged in with correct credentials.	Logged in via valid user credentials.	**Pass**
3	Click on SpO2 reading card	Should be redirected to different page to view SpO2 reading.	SpO2 reading is seen.	**Pass**
4	Click on delete button	Should delete the previous readings of the vitals.	Deletes the older readings of the vitals.	**Pass**
5	Click on Blood Pressure card	Should be redirected to different page to view blood pressure.	Blood pressure reading is seen.	**Pass**
6	Click on Temperature card	Should be redirected to different page to view temperature reading.	Temperature measurement is seen.	**Pass**
7	Click on Pulse rate card	Should be redirected to different page to view pulse rate.	Heart rate reading is seen.	**Pass**

8	Click on User icon	The user should be redirected to user profile page.	Redirected to user profile page.	**Pass**
9	Click on Delete button	The user should be able to delete their account.	Account is deleted when clicked on delete button.	**Pass**
10	Click on about icon	The user should be redirected to about us page.	Redirected to about us page.	**Pass**
11	Click on contact us icon	The user should be redirected to contact us page.	Redirected to contact us page.	**Pass**
12	Click on logout icon	The app should navigate back to login screen.	Redirected to login page.	**Pass**
13	Click on Track Location	The user is navigated to device location tracking page.	Redirected to device tracking page.	**Pass**
14	Click on location history	All the previous device location should be seen.	Previous device location was seen.	**Pass**

Table 3: Product Evaluation for Smart Vitals Mobile Application

CHAPTER 9: PROJECT EVALUATION

9.1 Task Sheet



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		1. Start Project	109 days	Tue 12/17/24	Sun 5/11/25		
2		**2. Initiation**	**17 days**	**Tue 12/17/24**	**Mon 1/6/25**		**Laptop[1 Hardware],MS Word[1 Softwa**
3		2.1 Research on the module	2 days	Tue 12/17/24	Wed 12/18/24		Laptop[1 Hardware],MS Word[1 Softwar
4		2.2 Research on Project title	5 days	Wed 12/18/24	Tue 12/24/24		Laptop[1 Hardware],MS Word[1 Softwar
5		2.3 Review of research article	3 days	Thu 12/26/24	Sat 12/28/24		Laptop[1 Hardware],MS Word[1 Softwar
6		2.4 Finalize Project title	3 days	Mon 12/30/24	Wed 1/1/25	5	Laptop[1 Hardware],MS Word[1 Softwar
7		2.5 Title Submission (Project Specification and Risk Register)	4 days	Thu 1/2/25	Mon 1/6/25	6	Laptop[1 Hardware],MS Word[1 Software],Notebook[1 Hardware],Pen[1
8		**3 .Planning**	**16 days**	**Mon 1/6/25**	**Fri 1/24/25**		**Laptop[1 Hardware],MS Project[1 Softw**
9		3.1 Schedule planning by breakingdown the project	4 days	Mon 1/6/25	Thu 1/9/25		Laptop[1 Hardware],MS Word[1 Softwar
10		3.2 Choose technologies to be used	4 days	Fri 1/10/25	Tue 1/14/25	9	Laptop[1 Hardware],MS Word[1 Softwar
11		3.3 Identify hardware and system requirements with their availability and budget allocation	2 days	Wed 1/15/25	Thu 1/16/25	10	Laptop[1 Hardware],MS Word[1 Software],Notebook[1 Hardware],Pen[1
12		3.4 Research on necessary tutorials, courses, or guides for tools and technologies to be used	4 days	Fri 1/17/25	Wed 1/22/25	11	Laptop[1 Hardware],MS Word[1 Software],Notebook[1 Hardware],Pen[1
13		3.5 Evaluate risks and their potential impacts	3 days	Wed 1/22/25	Fri 1/24/25		Laptop[1 Hardware],MS Word[1 Softwar
14		**4 Research On project**	**17 days**	**Fri 1/24/25**	**Sun 2/16/25**		**Laptop[1 Hardware],MS Project[1 Softw**
15		4.1 Research on IoT based projects related to my project	5 days	Fri 1/24/25	Thu 1/30/25		Laptop[1 Hardware],MS Word[1 Softwar
16		4.2 Research on relevant articles	3 days	Fri 1/31/25	Tue 2/4/25		Laptop[1 Hardware],MS Word[1 Softwar
17		4.3 Review different books based on IoT healthcare system	4 days	Tue 2/4/25	Fri 2/7/25		Laptop[1 Hardware],MS Word[1 Software],Notebook[1 Hardware],Pen[1
18		4.4 Understanding the working mechanism of Arduino and sensors involved	4 days	Fri 2/7/25	Wed 2/12/25		Laptop[1 Hardware],MS Word[1 Software],Notebook[1 Hardware],Pen[1
19		4.5 Finalize of Ethical Consent form	3 days	Wed 2/12/25	Fri 2/14/25		Laptop[1 Hardware],MS Word[1 Softwar
20		**5. Training and Implementation**	**47 days**	**Sun 2/16/25**	**Mon 4/21/25**		**MAX30100 Sensor[1 Hardware],3D prin**
21		5.1 Courses and Tutorials on Arduino, Sensors, and Health Monitoring	10 days	Sun 2/16/25	Thu 2/27/25		Notebook[1 Hardware], MAX30100 Sensor[1 Hardware],3D printed Box[1
22		5.2 Enorll in online classes for flutter and dart	6 days	Fri 2/28/25	Fri 3/7/25		Notebook[1 Hardware], MAX30100 Sens
23		5.3 Set up the development environment for C/C++	1 day	Sat 3/8/25	Sat 3/8/25		Notebook[1 Hardware], MAX30100 Sens
24		5.4 Commence writing and implementing the project's code	8 days	Sun 3/9/25	Tue 3/18/25		Notebook[1 Hardware], MAX30100 Sens
25		5.5 Initial testing of Sensor integration with ESP32	2 days	Wed 3/19/25	Thu 3/20/25		Notebook[1 Hardware], MAX30100 Sens
26		5.6 Integrating custom algorithms and code with the Arduino platform for seamless functionality	8 days	Fri 3/21/25	Tue 4/1/25		Notebook[1 Hardware], MAX30100 Sensor[1 Hardware],3D printed Box[1
27		5.7 Finalize Work In Progress with presentation	3 days	Tue 4/1/25	Thu 4/3/25		Notebook[1 Hardware], MAX30100 Sens
28		5.8 Integrating notification system	2 days	Fri 4/4/25	Mon 4/7/25		Notebook[1 Hardware], MAX30100 Sens
29		5.9 Develop Mobile Application using flutter	8 days	Sat 3/8/25	Tue 3/18/25		Notebook[1 Hardware], MAX30100 Sens
30		5.10 Design and Develop PCB and 3D casing with its integration	3 days	Fri 4/18/25	Tue 4/22/25		Notebook[1 Hardware], MAX30100 Sens
31		**6. Testing and Evaluating**	**10 days**	**Tue 4/22/25**	**Mon 5/5/25**		Notebook[1 Hardware], MAX30100 Sens
32		6.1 System Integration Testing	3 days	Tue 4/22/25	Thu 4/24/25		Notebook[1 Hardware], MAX30100 Sens
33		6.2 Functional Testing	3 days	Fri 4/25/25	Tue 4/29/25		Notebook[1 Hardware], MAX30100 Sens
34		6.3 Performance Testing	4 days	Wed 4/30/25	Mon 5/5/25		Notebook[1 Hardware], MAX30100 Sens
35		**7. Closing**	**7 days**	**Mon 5/5/25**	**Sun 5/11/25**		**3D printed Box[1 Hardware],Laptop[1 Ha**
36		7.1 Final Product Submission and Presentation	1 day	Mon 5/5/25	Mon 5/5/25		3D printed Box[1 Hardware],Laptop[1 Ha
37		7.2 Final Report Submission	1 day	Sun 5/11/25	Sun 5/11/25		3D printed Box[1 Hardware],Laptop[1 Ha

Figure 30: Task Sheet

The above figure represents the complete flow of how the project was successfully concluded by following a planned task sheet.

9.2 Gnatt Chart

Figure 31: Gnatt Chart

The Gnatt Chart from the above figure is a visual display of the project flow, which tracks the tasks from start to the end keeping track of the project development along different dates.

9.3 Timeline

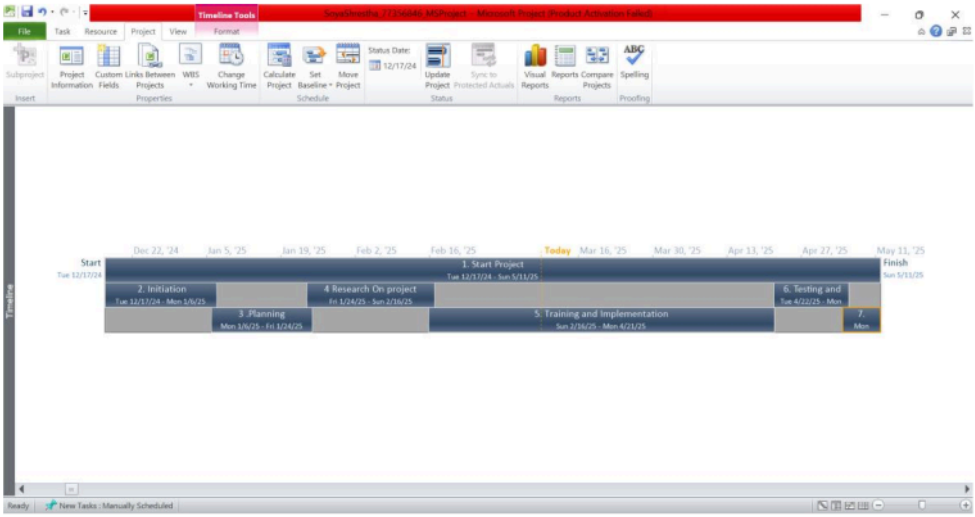
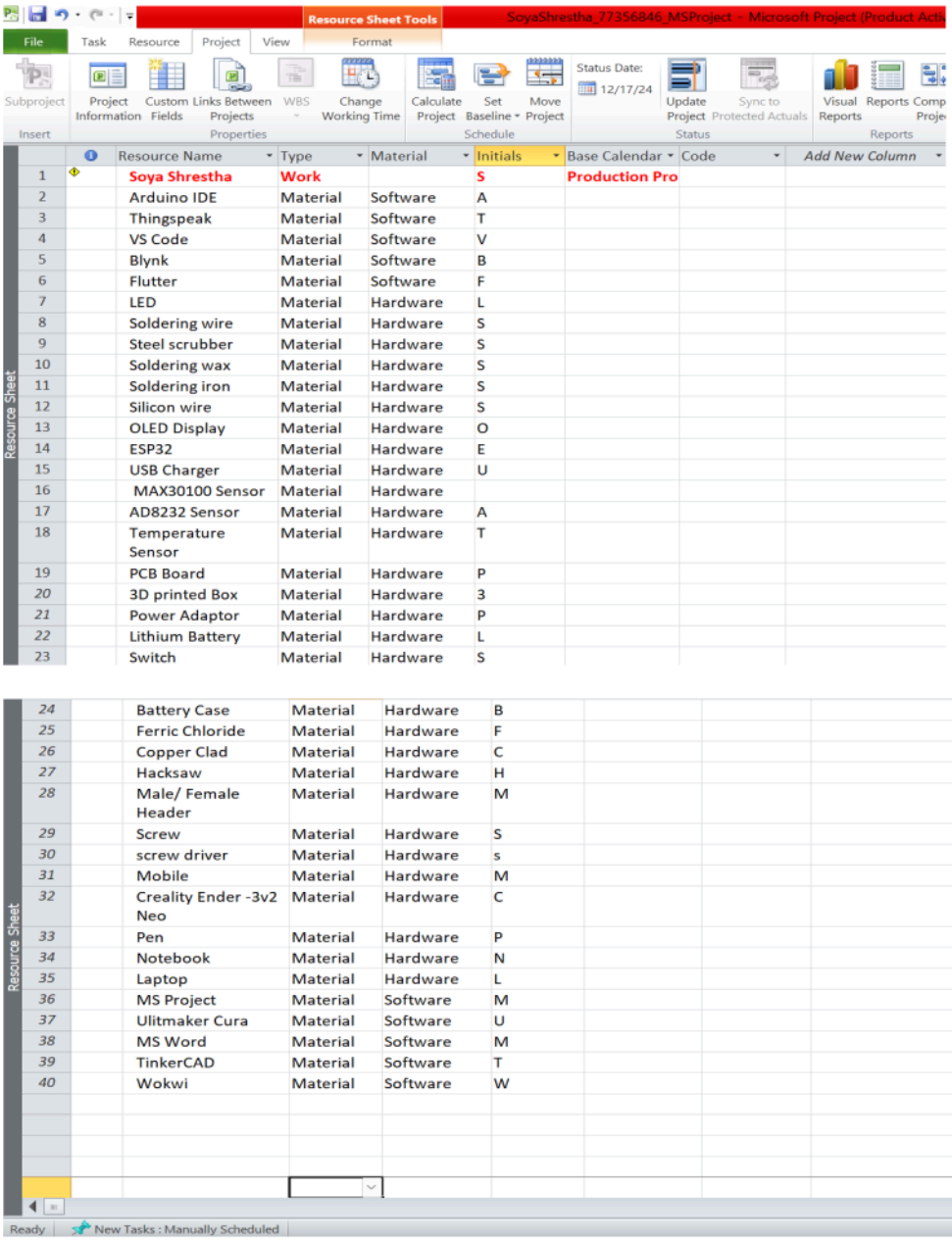


Figure 32: Project Timeline

Key stages of the project are shown in the MS Project Timeline, with initiation phase starting at 12/17/2024, followed by Planning phase, Research phase, Training and implementation phase, Testing and Evaluation phase with Closing phase marking the end of the project at 05/11/2025.

9.4 Resource Sheet



	Resource Name	Type	Material	Initials	Base Calendar	Code	Add New Column
1	Soya Shrestha	Work		S	Production Pro		
2	Arduino IDE	Material	Software	A			
3	Thingspeak	Material	Software	T			
4	VS Code	Material	Software	V			
5	Blynk	Material	Software	B			
6	Flutter	Material	Software	F			
7	LED	Material	Hardware	L			
8	Soldering wire	Material	Hardware	S			
9	Steel scrubber	Material	Hardware	S			
10	Soldering wax	Material	Hardware	S			
11	Soldering iron	Material	Hardware	S			
12	Silicon wire	Material	Hardware	S			
13	OLED Display	Material	Hardware	O			
14	ESP32	Material	Hardware	E			
15	USB Charger	Material	Hardware	U			
16	MAX30100 Sensor	Material	Hardware				
17	AD8232 Sensor	Material	Hardware	A			
18	Temperature Sensor	Material	Hardware	T			
19	PCB Board	Material	Hardware	P			
20	3D printed Box	Material	Hardware	3			
21	Power Adaptor	Material	Hardware	P			
22	Lithium Battery	Material	Hardware	L			
23	Switch	Material	Hardware	S			
24	Battery Case	Material	Hardware	B			
25	Ferric Chloride	Material	Hardware	F			
26	Copper Clad	Material	Hardware	C			
27	Hacksaw	Material	Hardware	H			
28	Male/ Female Header	Material	Hardware	M			
29	Screw	Material	Hardware	S			
30	screw driver	Material	Hardware	s			
31	Mobile	Material	Hardware	M			
32	Creality Ender -3v2 Neo	Material	Hardware	C			
33	Pen	Material	Hardware	P			
34	Notebook	Material	Hardware	N			
35	Laptop	Material	Hardware	L			
36	MS Project	Material	Software	M			
37	Ulitmaker Cura	Material	Software	U			
38	MS Word	Material	Software	M			
39	TinkerCAD	Material	Software	T			
40	Wokwi	Material	Software	W			

Figure 33: Resource Sheet

The above figure consists of all the tools and technologies (Hardware and Software) used, which are essential to complete the project.

9.5 GitHub

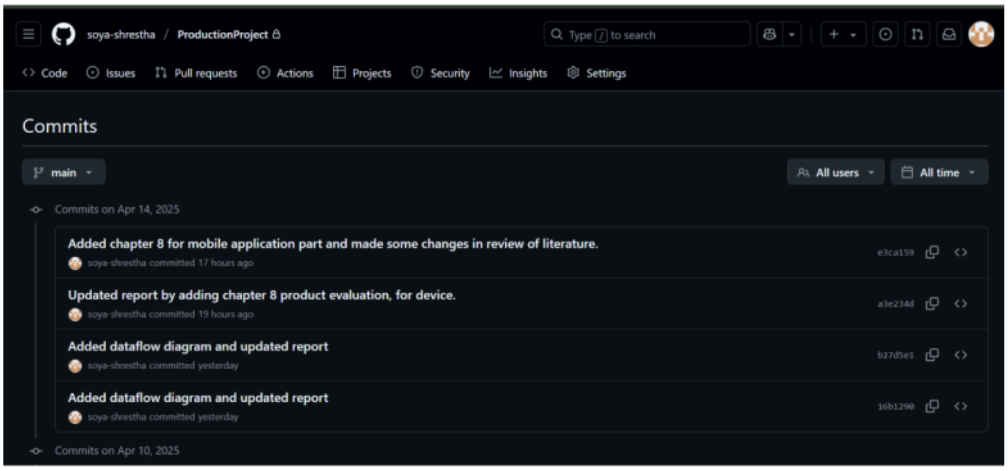


Figure 34: GitHub

GitHub was used to manage the codes and files in a systematic way ensuring regular commits to see the progression of the project. The git hub link which shows the progressions of the project is given below:

Git Hub Link: *https://github.com/soya-shrestha/ProductionProject*

9.6 Entity Relationship Diagram (ERD) and Composite ERD

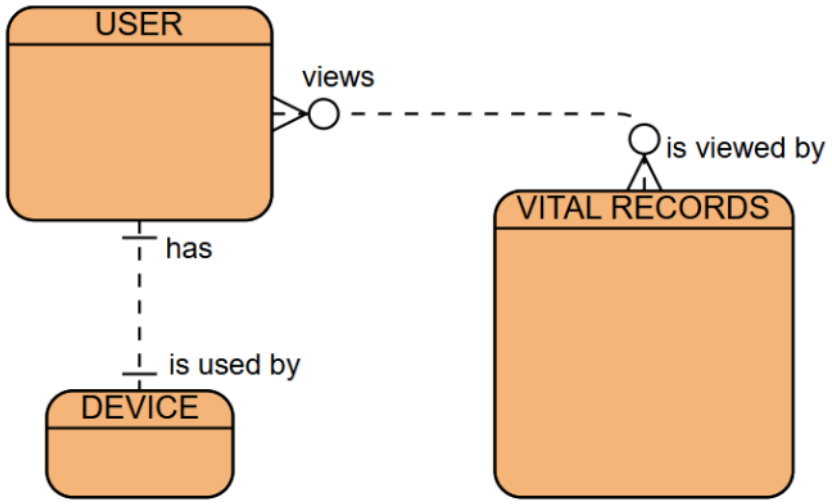


Figure 35: Entity Relationship Diagram (ERD)

The above ERD demonstrates the basic structure of how user is associated with the device and their vital records. A device is used by a user to get their vital readings. The generated vital records can be viewed by the users.

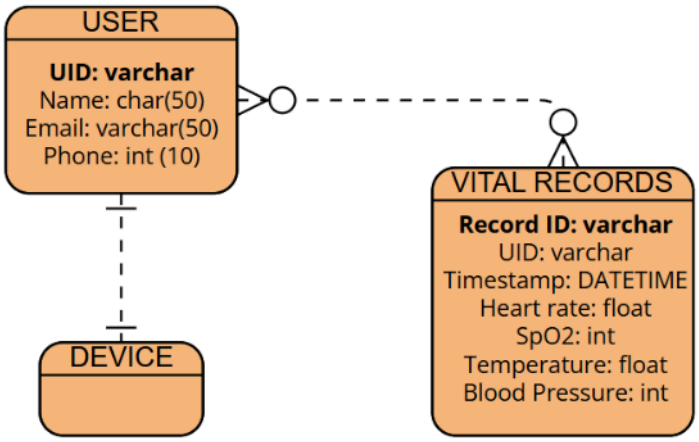


Figure 36: Composite ERD

The above figure is the Composite ERD of Smart Vitals, which shows a detailed relation between entity and their attributes; each user has their own unique UID with personal data. Vital records of each user are linked via UID (primary key) which is the foreign key to the vital records similarly Record ID is the primary key to the Vital Records entity. The entity “device” is linked with the entity “User” with one – one relationship (1:1). Similarly, “User” entity is linked with entity “Vital Records” with many – many relationship (M: N).

9.7 UML

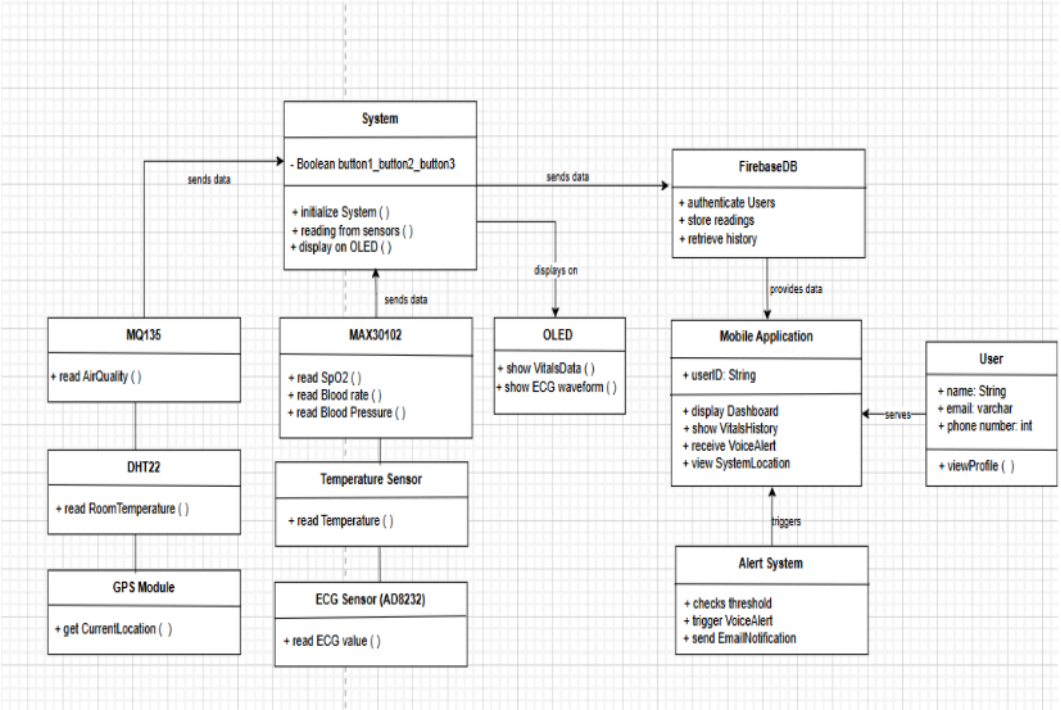


Figure 37: UML for Smart Vitals

The UML diagram below displays the structure of the Smart Vitals. The system module communicated with multiple sensors to measure health vitals in order to retrieve data via button press and environment conditions. The data retrieved via sensor are displayed on the OLED, concurrently sending data to Firebase database. The Firebase DB is then used to display data and history via mobile application linked to particular user. Additionally, alert via voice and email notification is triggered when reading is below threshold.

9.8 Requirement Catalogue

The tables below show the functional requirement and non – functional requirements with the MoSCoW method.

Functional Requirements	MoSCoW
For flawless interaction the device must be linked with the Wi-Fi.	**M**
Buttons to start and stop the measurement of vitals through various sensors.	**M**
OLED display to get easy and quick local access to the real-time health vitals with necessary message for user convenience.	**M**
Mobile application should be able to display vital readings.	**S**
Voice alert via mobile application for any irregularities in the readings.	**S**
Alert via email for unusual readings with device location.	**S**
The mobile application could be able to get knowledge on air quality and room temperature and device location coordinates (latitude and longitude).	**C**
User can save the vitals history.	**C**
User can delete the vitals history if they want.	**C**
The user could delete their profile.	**C**
Users could get insights on the location of the device.	**C**
User could be able to view location history.	**C**
If the OLED screes freeze the user can press the reset button to restart the micro controller.	**W**

Table 4: Functional Requirement for Smart Vitals

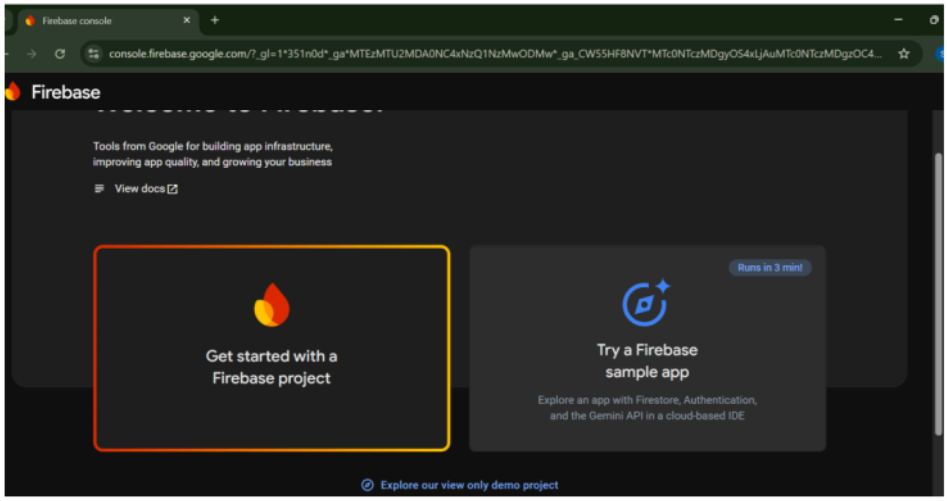
Non - Functional Requirements	MoSCoW
Smart Vitals must be convenient, compact and easy-to-use.	**M**
The OLED must display the measurements on the spot.	**M**
The battery operating the Smart Vitals should last for a day or 2.	**S**
Using the type B cable the Li – ion batteries should be charged.	**S**
Further additional of the sensor could be supported by the device.	**C**

Table 5: Non - functional Requirement for Smart Vitals

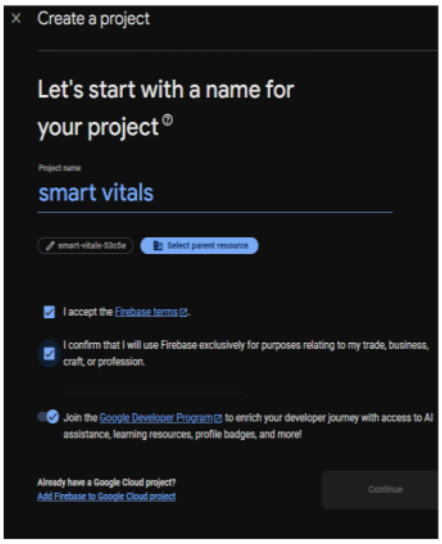
9.9 Installation Guidelines

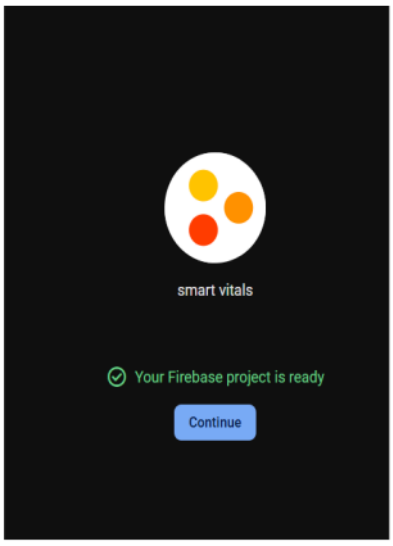
9.9.1 Firebase Installation Guidelines

Step 1: Before beginning with the installation of firebase package, it is necessary to create project through firebase console by clicking on the Get started with a Firebase project.

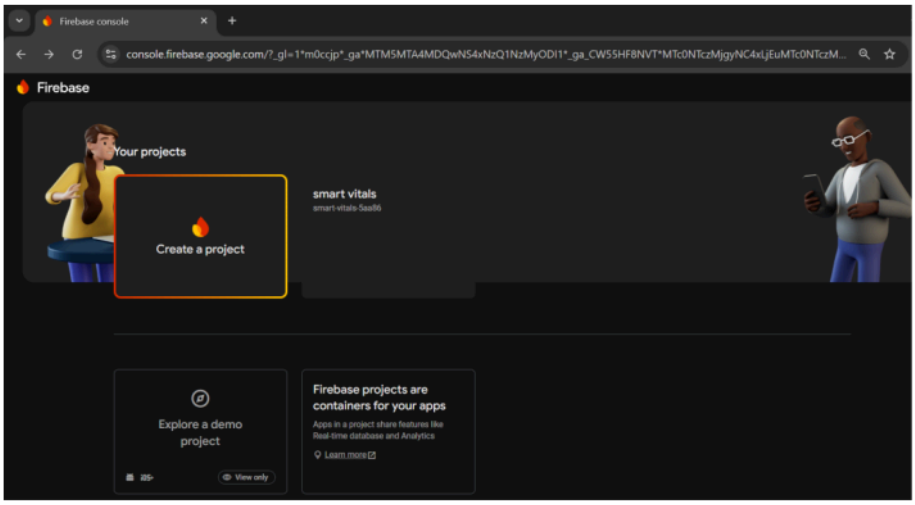


Step 2: Give an appropriate name for your project, and click on continue by agreeing to the terms and conditions which will then show the Firebase project being ready.



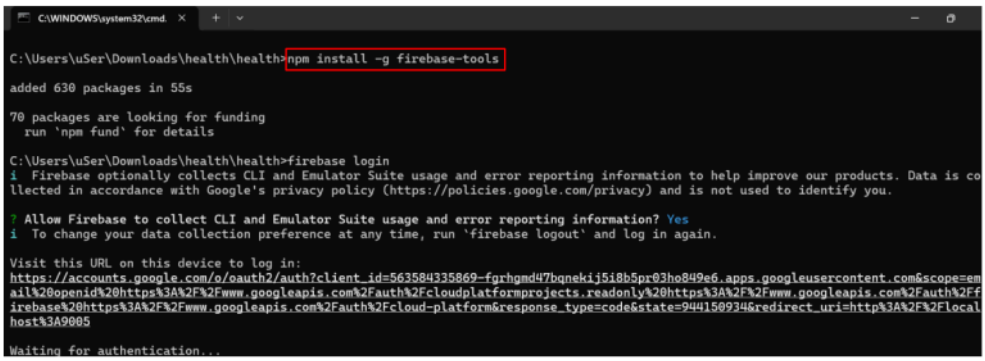


Step3: All the created project can be seen in the firebase console.



Step 4: Once the setup for the firebase console has been done, the installation process can begin.
Open cmd on your laptop, and enter “npm install firebase”, this command should install necessary firebase SDK packages.

Step 5: Next enter “npm install -g firebase-tools” which installs Firebase CLI, which allows us to run global different commands.



Step 6: The command firebase init allows to set up the project folder with the Firebase which allows the user to build, run and deploy the app.

Step 7: Then, database schemas, was set up with data connection and then Dart SDK was created which enabled local building and evaluation.

```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to
confirm your choices. Data Connect: Set up a Firebase Data Connect service, Firestore: Configure security rules and
indexes files for Firestore, Realtime Database: Configure a security rules file for Realtime Database and (optionally)
provision default instance

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: smart-vitals-d11dd (Smart Vitals )
i  Using project smart-vitals-d11dd (Smart Vitals )

=== Dataconnect Setup
i  dataconnect: ensuring required API sqladmin.googleapis.com is enabled...
!  dataconnect: missing required API sqladmin.googleapis.com. Enabling now...
+  dataconnect: required API sqladmin.googleapis.com is enabled
+  Wrote dataconnect\dataconnect.yaml
+  Wrote dataconnect\schema\schema.gql
+  Wrote dataconnect\connector\connector.yaml
+  Wrote dataconnect\connector\queries.gql
+  Wrote dataconnect\connector\mutations.gql
+  Detected multiple app platforms in directory C:\Users\uSer\Downloads\health\health
? Which platform do you want to set up a generated SDK for? Flutter (Dart)
? Which connector do you want set up a generated SDK for? health/default
i  Writing your new SDK configuration to C:\Users\uSer\Downloads\health\health\dataconnect\connector/connector.yaml
? File dataconnect\connector\connector.yaml already exists. Overwrite? Yes
+  Wrote dataconnect\connector\connector.yaml
i  dataconnect: downloading dataconnect-emulator-2.1.0.exe...
Progress: ==========================================================================================> (100% of 27MB)
I0410 16:26:43.135051   18176 codegen.go:82] [connector "default" dartSdk] Generating sources into "C:\\Users\\uSer\\Downloads\\healt
h\\health\\dataconnect-generated\\dart\\default_connector"
```

Step 8: After that, Firebase indexes and rules which allows API of Realtime Database, creating a new Realtime instance for Database and selecting the location for database instance.

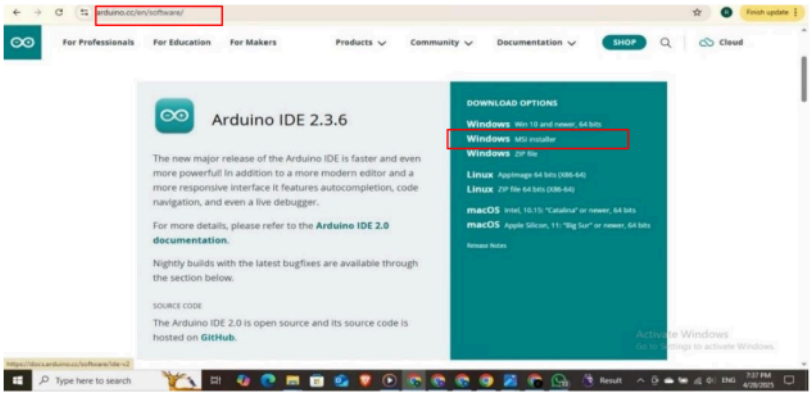
```
Firestore Security Rules allow you to define how and when to allow
requests. You can keep these rules in your project directory
and publish them with firebase deploy.

? What file should be used for Firestore Rules? firestore.rules

Firestore indexes allow you to perform complex queries while
maintaining performance that scales with the size of the result
set. You can keep index definitions in your project directory
and publish them with firebase deploy.

? What file should be used for Firestore indexes? firestore.indexes.json

=== Database Setup
i  database: ensuring required API firebasedatabase.googleapis.com is enabled...
!  database: missing required API firebasedatabase.googleapis.com. Enabling now...
+  database: required API firebasedatabase.googleapis.com is enabled

? It seems like you haven't initialized Realtime Database in your project yet. Do you want to set it up? Yes
? Please choose the location for your default Realtime Database instance: us-central1
/ Creating your default Realtime Database instance: smart-vitals-d11dd-default-rtdb

Firebase Realtime Database Security Rules allow you to define how your data should be
```

Step 9: Then the firebase setup was completed after selecting the file for security rules for Realtime Database.

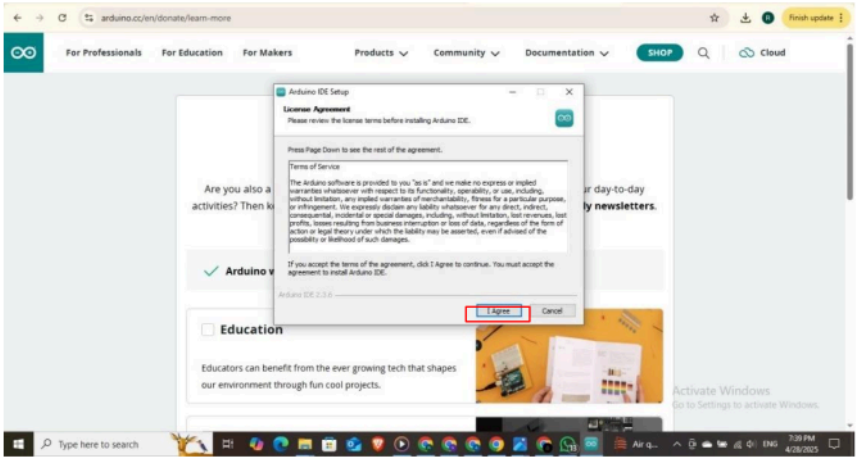
```
? What file should be used for Realtime Database Security Rules? database.rules.json
+  Database Rules for smart-vitals-d11dd-default-rtdb have been written to database.rules.json.
Future modifications to database.rules.json will update Realtime Database Security Rules when you run
firebase deploy.

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

+  Firebase initialization complete!
```

9.9.2 Arduino IDE installation guidelines

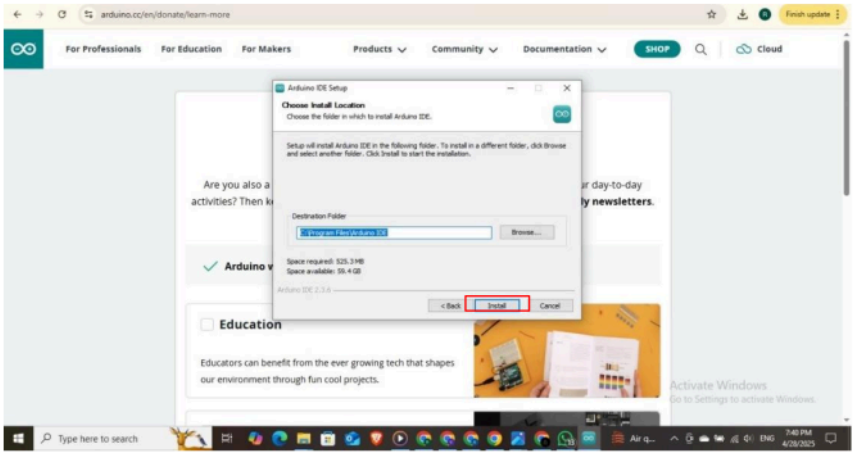
Step 1: To install Arduino IDE, we should go to: https://www.arduino.cc/en/software/ where the Operating system should be selected.



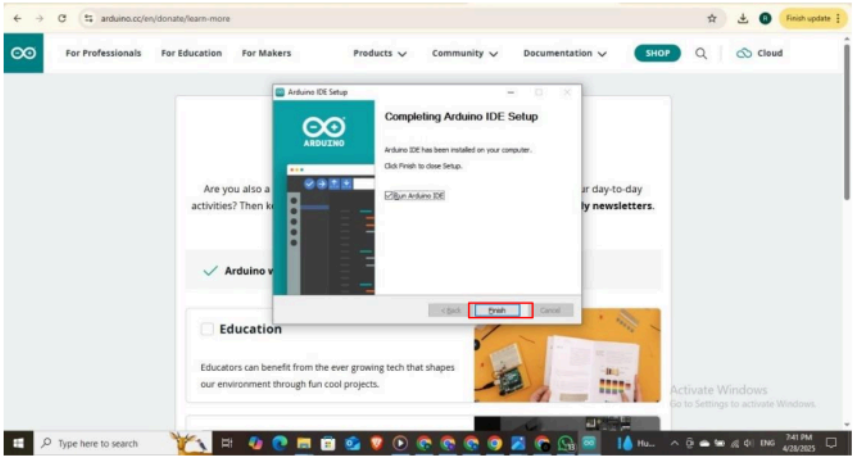
Step 2: Then, a License Agreement is seen where, click “I Agree”, which should start the installation.



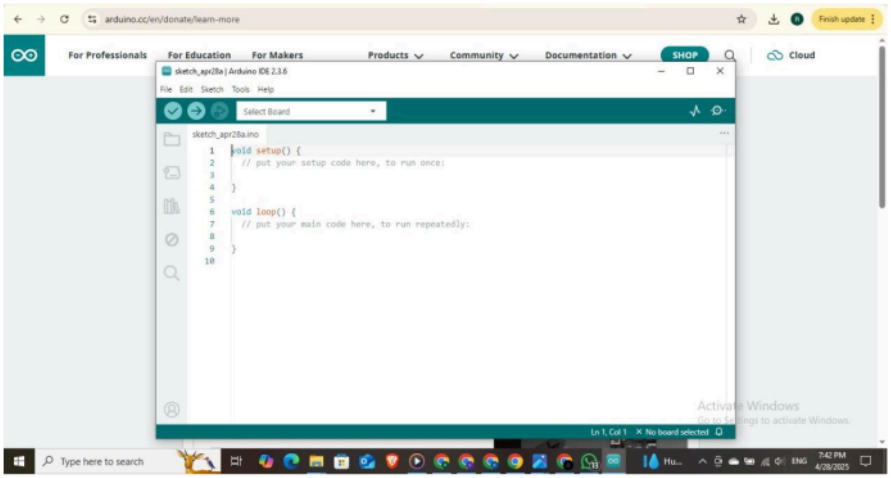
Step 3: After that, select the destination folder - where you want to keep your application. Then click on install which should start the installation.



Step 4: The setup is completed after the “Completing Arduino IDE Setup” screen is seen as below. Click on Finish to run Arduino IDE.

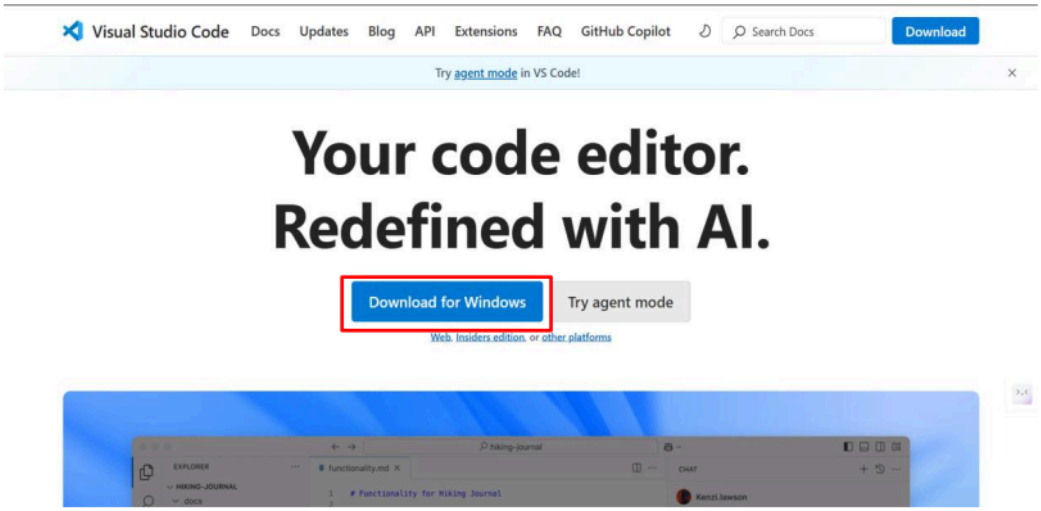


Step 5: As you can see, the Arduino IDE is setup and now you start coding.

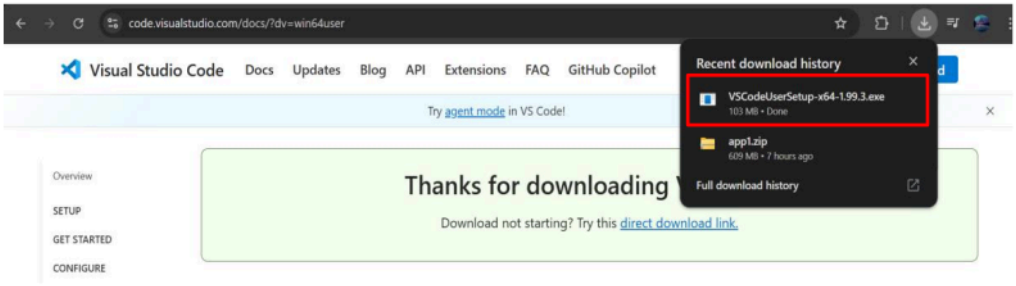


9.9.3 VS code installation guidelines

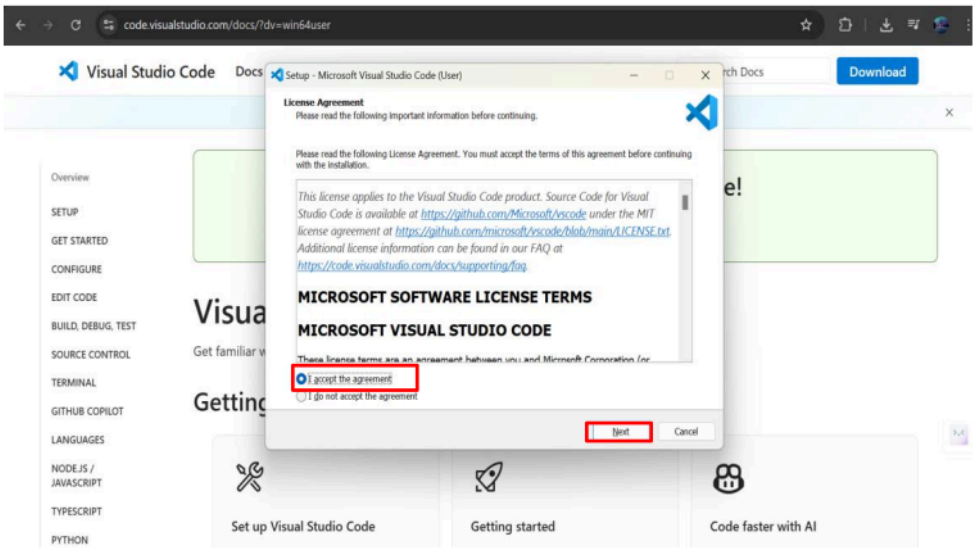
Step 1: Go *to https://code.visualstudio.com/*. Then click on Download for Windows if you are using Windows Operating System.



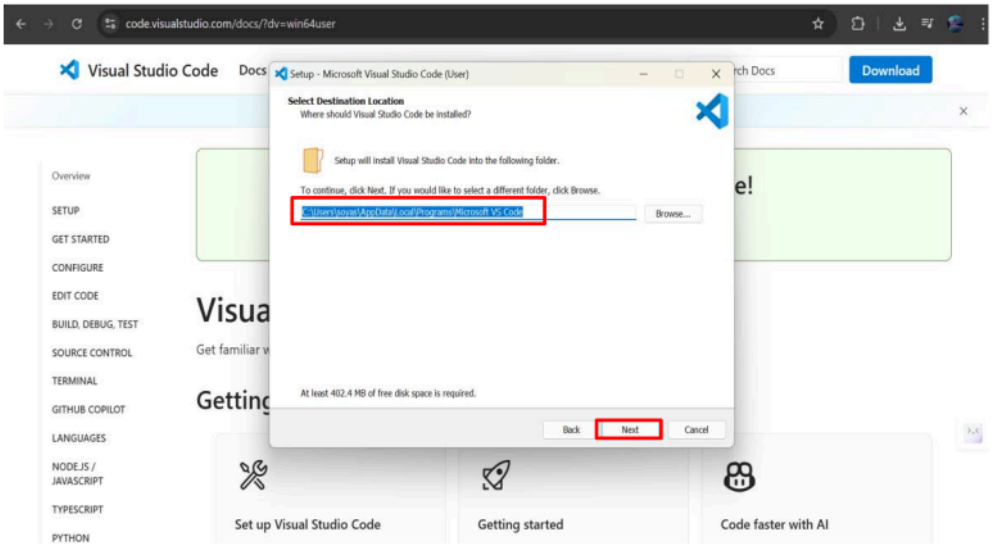
Step 2: The setup file should be downloaded as shown in the picture below, which when clicked should start the installation process of VS code.



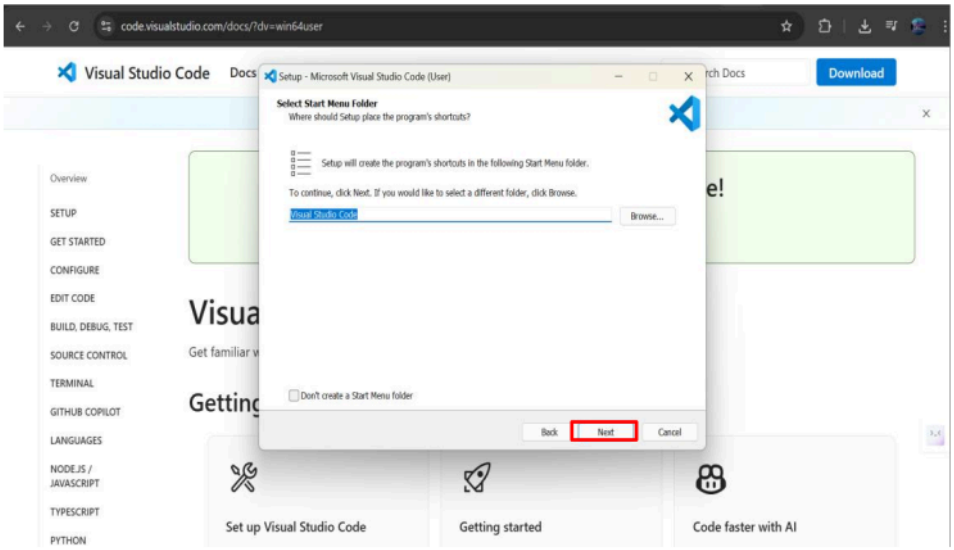
Step 3: Now click on “I accept the agreement” radio button and then on Next button to continue.



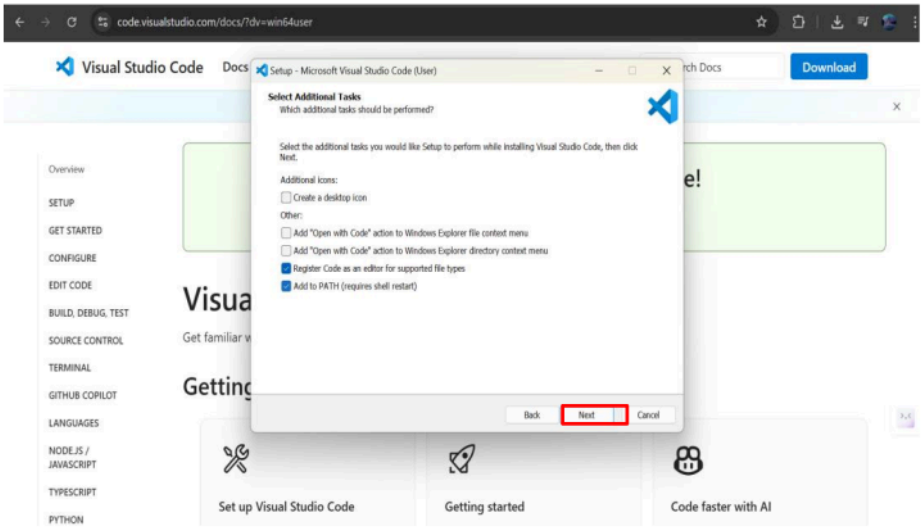
Step 4: To continue, Click on Next button or select the file destination according to your desire and then click on Next button.



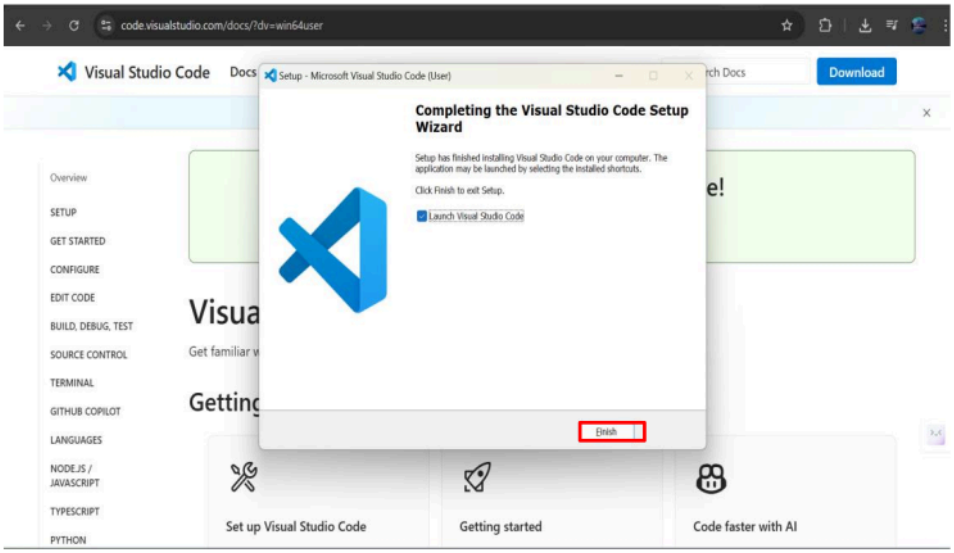
Step 5: Click on the Next button to continue the process.



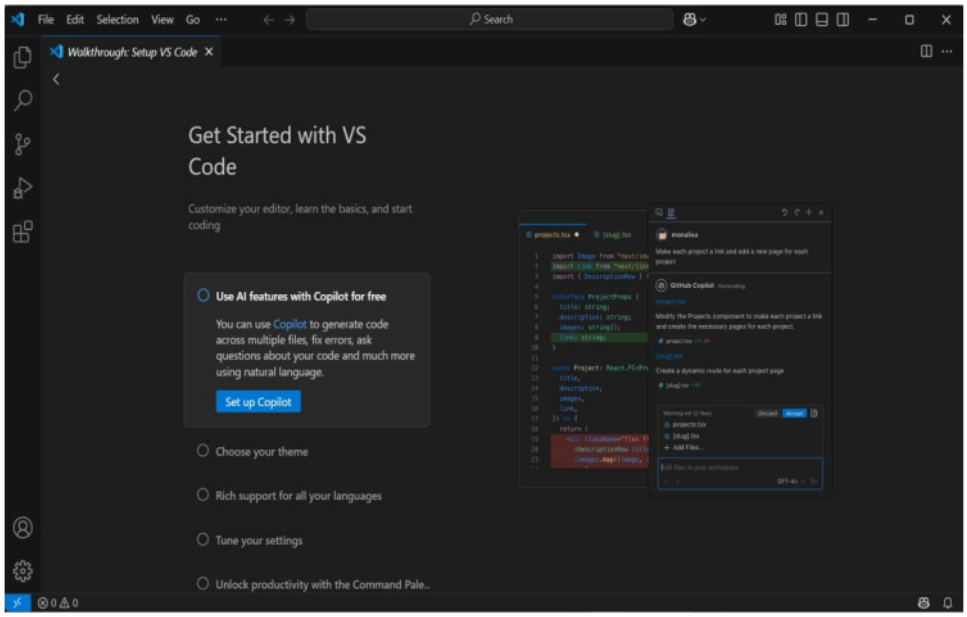
Step 6: Leave the default selection or select the additional tasks and then click on Next button.



Step 7: The setup for Visual Studio is now completed, click on the Finish button to get started with VS code.



Step 8: After clicking on the finish button, VS code is successfully installed. BY clicking on file -> new file, you can start with the coding in the VS code.



9.10 Use Case Diagram

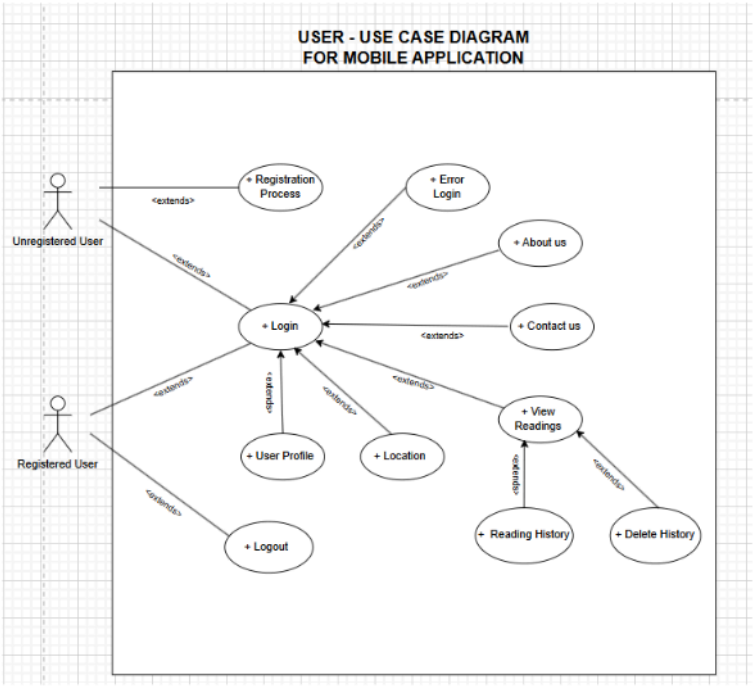


Figure 38: User Use Case Diagram for Smart Vitals (1)

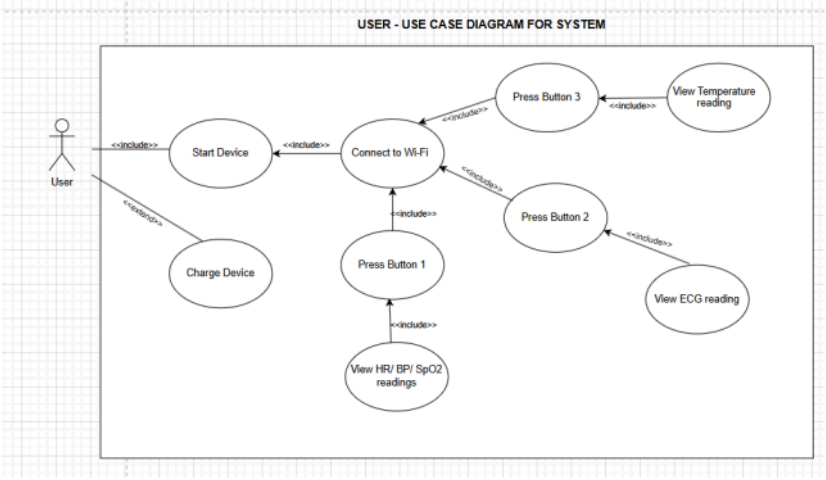


Figure 39: User Use Case Diagram for Smart Vitals (2)

The above use case diagram represents how users are associated with the Smart Vitals system and mobile application giving a perspective of the operations inside the system boundary.

9.11 Circuit Diagram

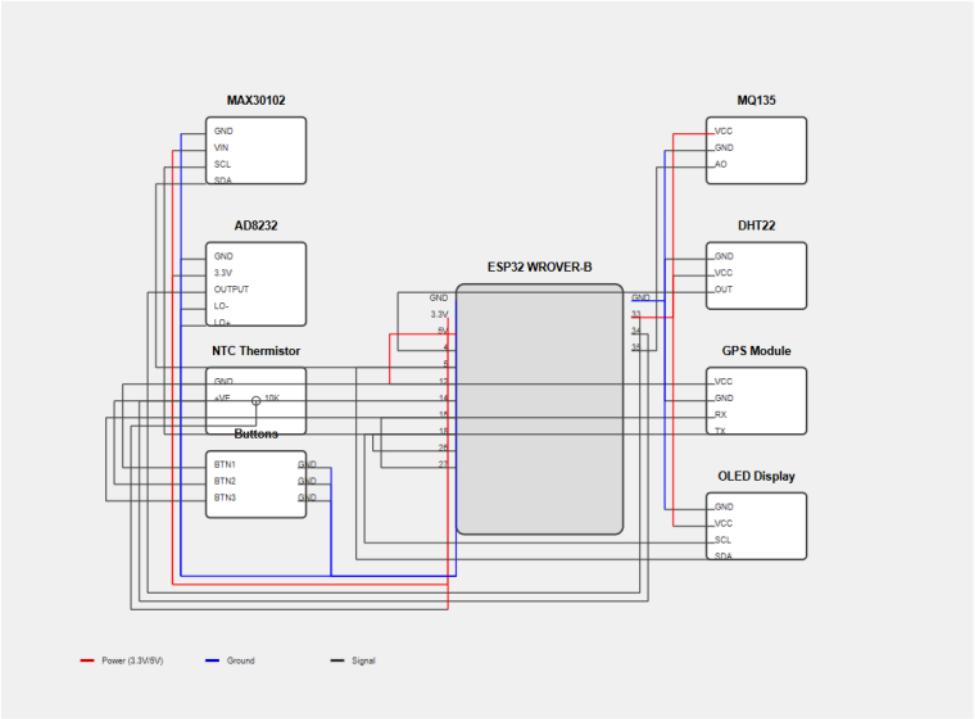


Figure 40: Circuit Diagram for Smart Vitals

The above circuit diagram represents how each component is associated with the micro controller (ESP 32 Wrover Type B). The positive power supply is represented by the red color wire connections; blue is represented by ground wire (negative power supply) similarly black wire connection is the signal or output connection.

CHAPTER 10: SUMMARY AND CONCLUSION

Smart Vitals was developed to help people get insights on their health conditions. The main focus of the project was on improving health management and easy accessibility to the vitals. It is portable vitals monitoring system that helps users to measure their vitals. The device was designed in such a way that the users can easily get their required measurements with the integration of buttons. If there is a stable internet connection the measurements can be accessible via mobile app however OLED has been included in the system for local access, making it easy and quick to gain readings of the vitals. Voice alert has been added to the mobile app making it a smart feature with live location of the device for helping the users to take instant action. As an additional feature for Smart Vitals, air quality and temperature monitoring has been added to get insights on the environment conditions. The app allows users to save the vital readings according to their need, which could be necessary for further examination allowing the doctors or caretakers to keep track and gain knowledge on their past health history. While testing the product, all the necessary and important features were found to be decent. After doing several testing it was found that the Temperature, using 10k NTC thermistor thermometer and SpO2 using MAX30102 was 99% accurate. While Heart rate, using MAX30102 was 95% accurate. Similarly, the blood pressure was 97% accurate. Further experiments were conducted to test the alert feature; it was found that voice alert was working just fine. Overall, the device performed fair enough, making it a viable product for commercial market on further research and development.

There are many existing products in the market that are somewhat similar to Smart Vitals, while it is still a work in progress device, it has some features that stands out. As per (Valiaugaitė, 2025), Withings BeamO is a portable 4 in 1 device that measures the temperature, blood oxygen level, heart activities and respiratory system. While BeamO is device specifically designed for user health monitoring, Smart Vitals not only measures users' vitals but also the environments conditions like air quality and temperature. The device is integrated with GPS module so that user can easily get access to the location in case of

emergency, with stable Wi-Fi connection. Smart Vitals can give users an estimated measurement for blood pressure while BeamO lacks the feature. BeamO is a high-end premium product which costs around $249.95 according to (Bouchard, 2024), which might not be economical device for the users who are looking for cost effective device, however smartvitals on the other hand is cost effective providing users with more affordable solution. SmartVitals is still a prototype device, additional sensors could be added for more health readings, whereas BeamO is fully developed product thus, additions of more sensors are hardly possible.

Smart Vitals is an example of outcome of technology and health together making it a prototype device for promoting those sectors. Since it is just at a development stage i.e. a work in progress model it could be made into a market-oriented product using more advance and top-tier sensors. To take Smart Vitals to that level several changes could be made. Inclusion of additional sensors such as Blood Glucose Sensor, Electromyography, Stress Level Sensors, etc. could be added for additional vital monitoring. Additionally, further advanced analysis of ECG could potentially be a good additional to the device. Besides that, in the device battery could be further optimized for enhanced battery life. Smart Vitals does have voice alert, but inclusion of real-time alert with notification could be helpful for users to gain knowledge on the users' irregularities on health vitals. While smartvitals has an accuracy rate of around 96%, it could be improved with the usage of different sensors, more in-depth study and proper calibrations. The usage of AI in the system would be a great way to continue the smart vital project in future as it would help users to predict the health conditions and analyze vital trends. As of now Smart Vitals operates only on English language, which could be upgraded where it supports multiple language according to user preference, allowing easy interaction with the device as well as the mobile application. Integration of multiple language in Smart Vitals could be the future enhancement ideas. While these are few examples of how smart vitals can be upgraded in the future, there are many other ways to enhance the system making it compatible with other exiting products in the market.

FinalYearReport_SoyaShrestha_77356846.pdf

ORIGINALITY REPORT

5%	2%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to The British College Student Paper	2%
2	ouci.dntb.gov.ua Internet Source	1%
3	eprints.utm.my Internet Source	<1%
4	docplayer.net Internet Source	<1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Softwarica College Of IT & E-Commerce Student Paper	<1%
7	www.economist.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off

Exclude bibliography On